IBM Research

Sieving for closest lattice vectors (with preprocessing)

Thijs Laarhoven

mail@thijs.com
http://www.thijs.com/

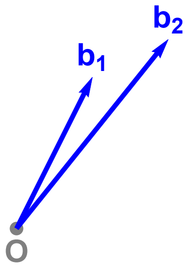
SAC 2016, St. John's (NL), Canada
(August 12, 2016)

Lattices

What is a lattice?

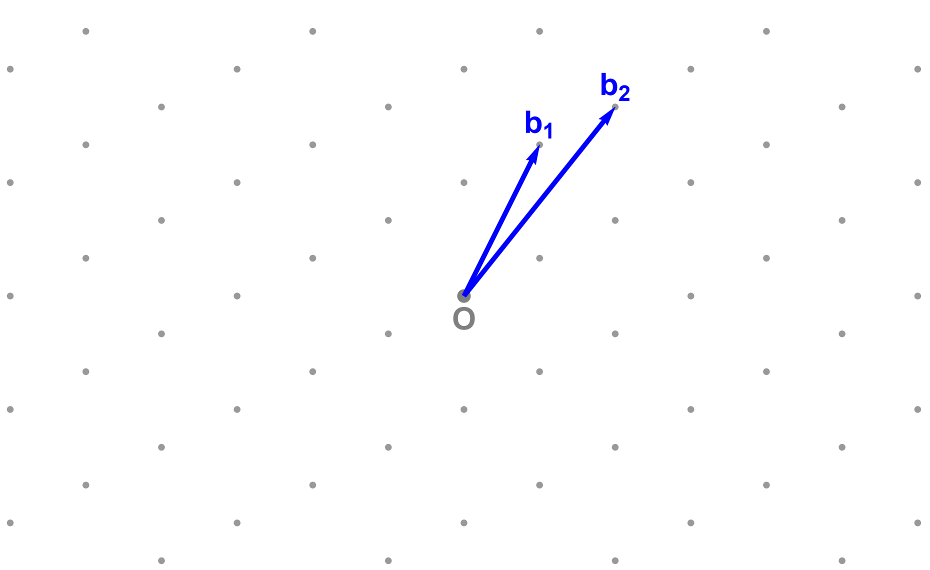
Lattices

What is a lattice?



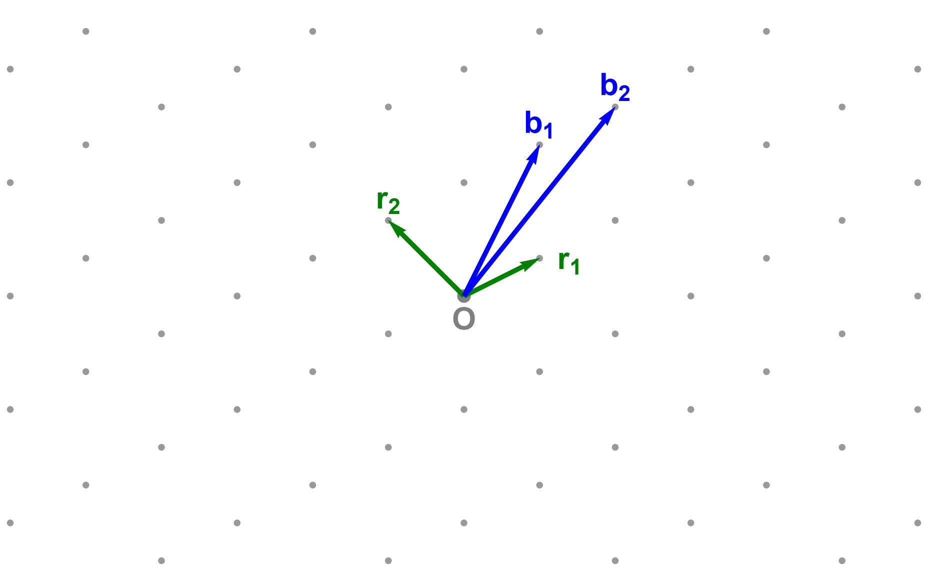
Lattices

What is a lattice?



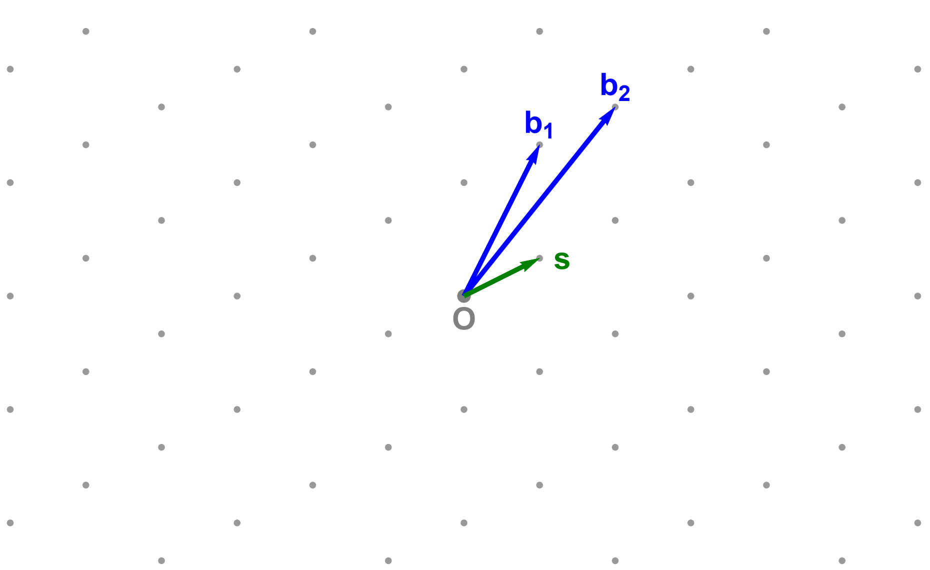
Lattices

Lattice basis reduction



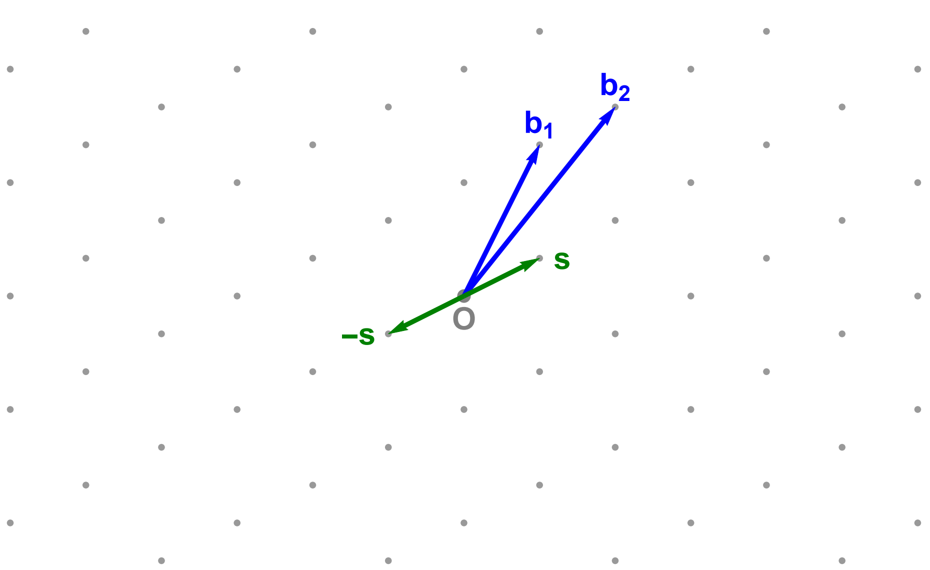
Lattices

Shortest Vector Problem (SVP)



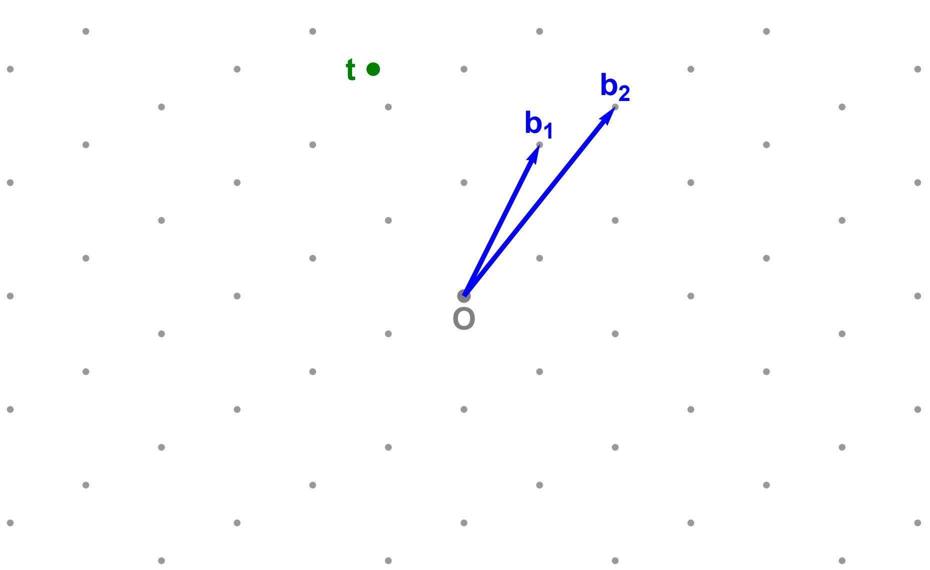
Lattices

Shortest Vector Problem (SVP)



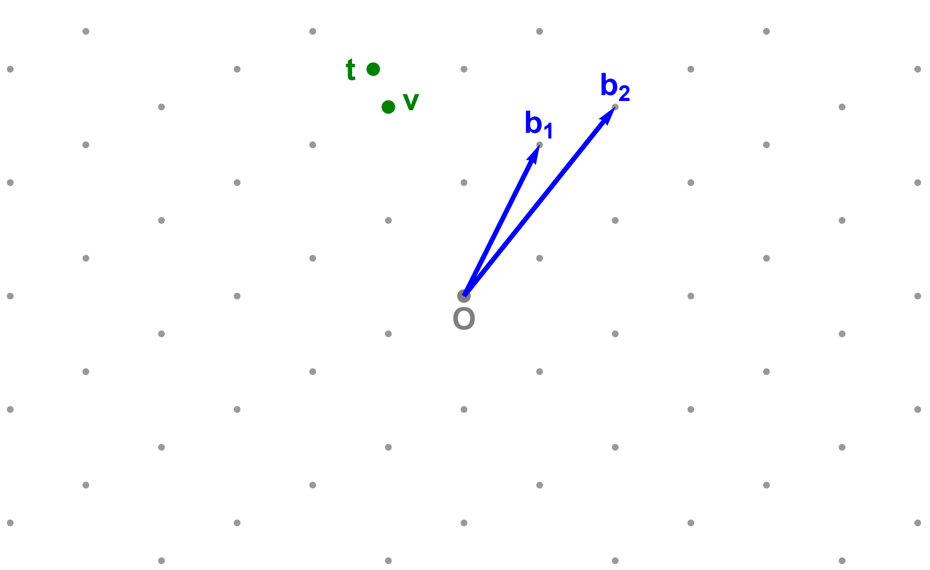
Lattices

Closest Vector Problem (CVP)



Lattices

Closest Vector Problem (CVP)



Outline

Sieving for SVP

Sieving for CVP

Sieving for CVPP

Conclusion

Outline

Sieving for SVP

Sieving for CVP

Sieving for CVPP

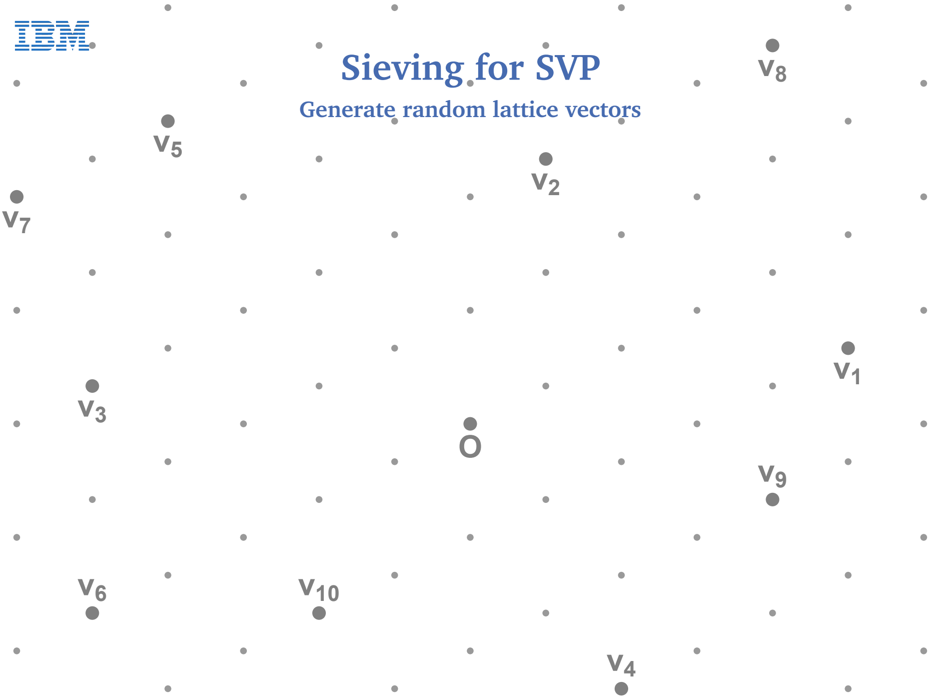
Conclusion

Sieving for SVP

Generate random lattice vectors

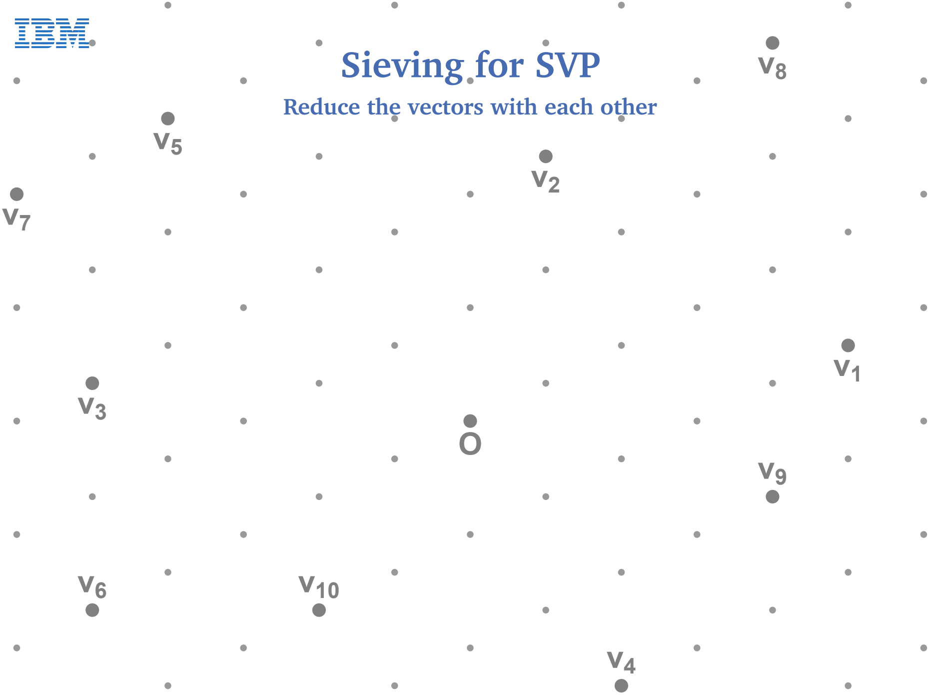
Sieving for SVP

Generate random lattice vectors



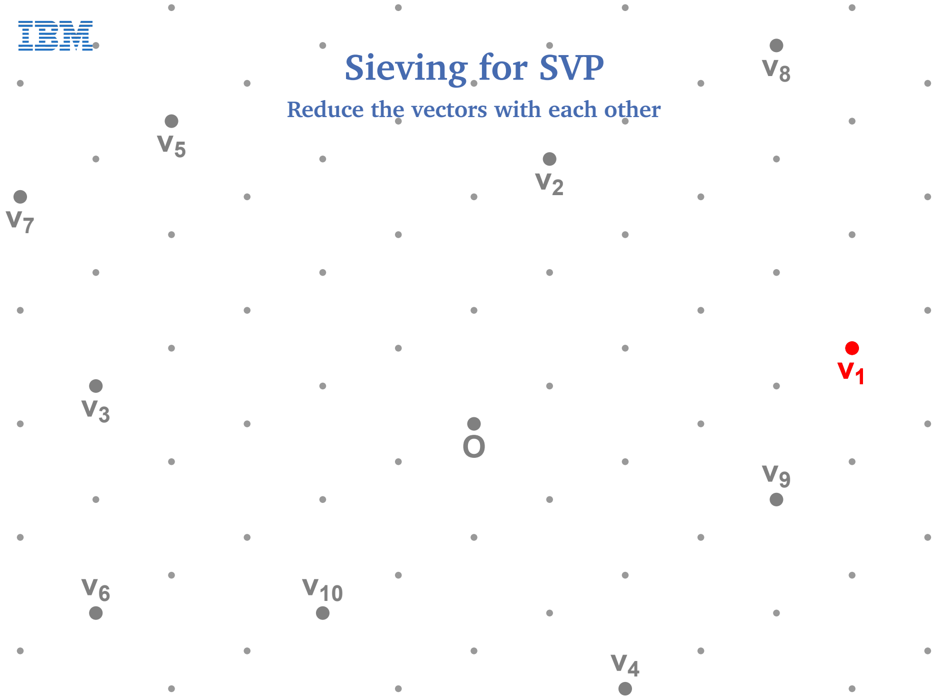
Sieving for SVP

Reduce the vectors with each other



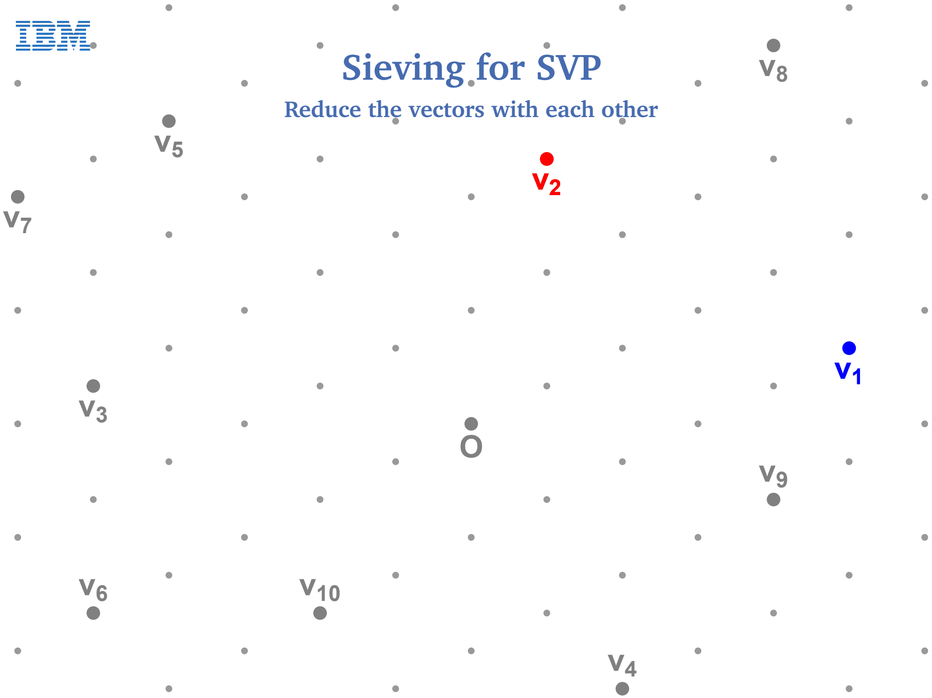
Sieving for SVP

Reduce the vectors with each other



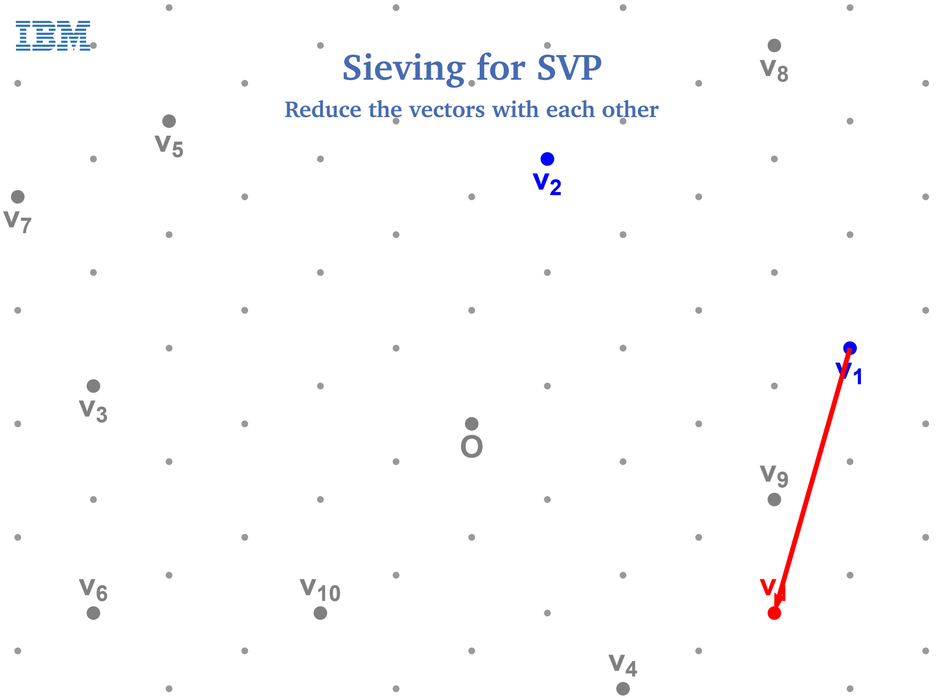
Sieving for SVP

Reduce the vectors with each other



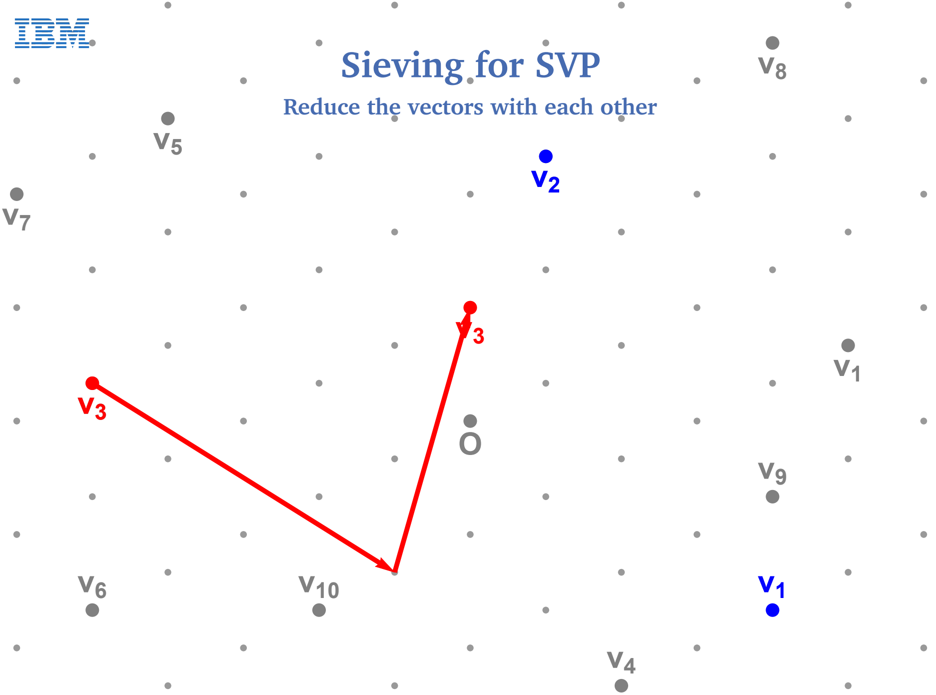
Sieving for SVP

Reduce the vectors with each other



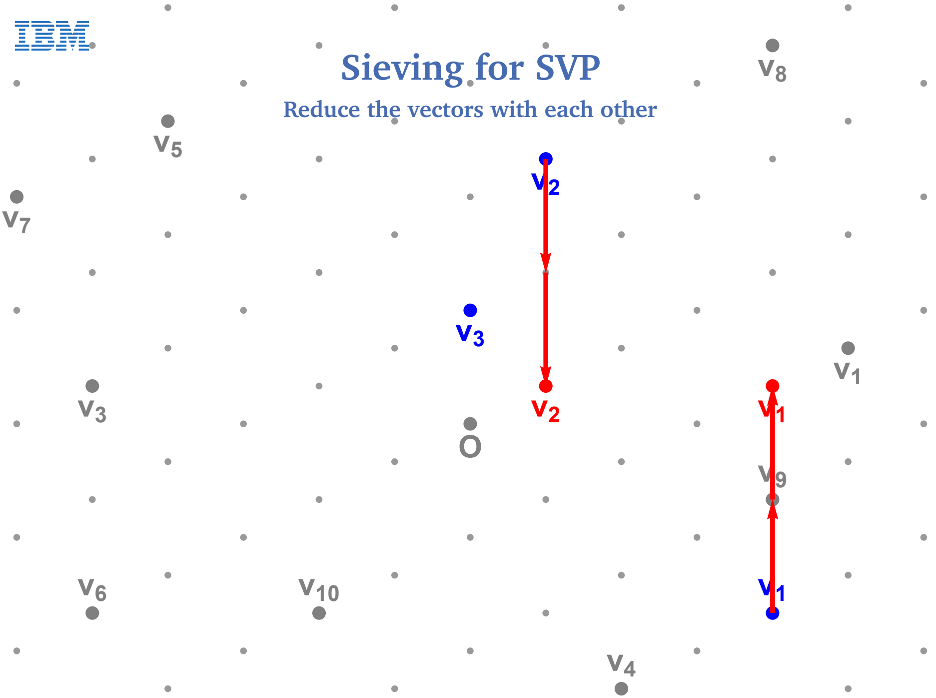
Sieving for SVP

Reduce the vectors with each other



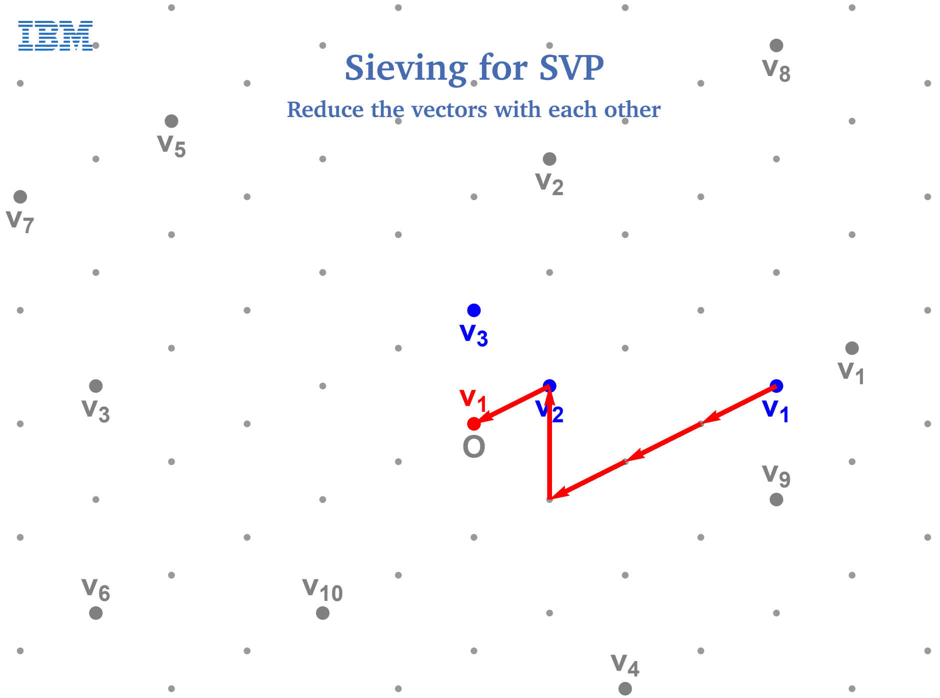
Sieving for SVP

Reduce the vectors with each other



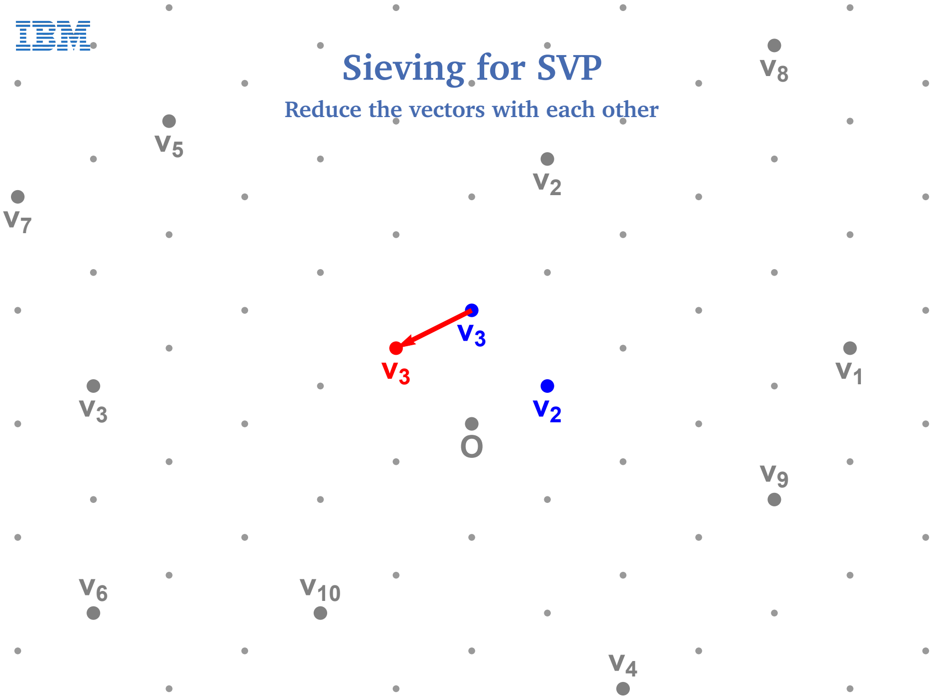
Sieving for SVP

Reduce the vectors with each other



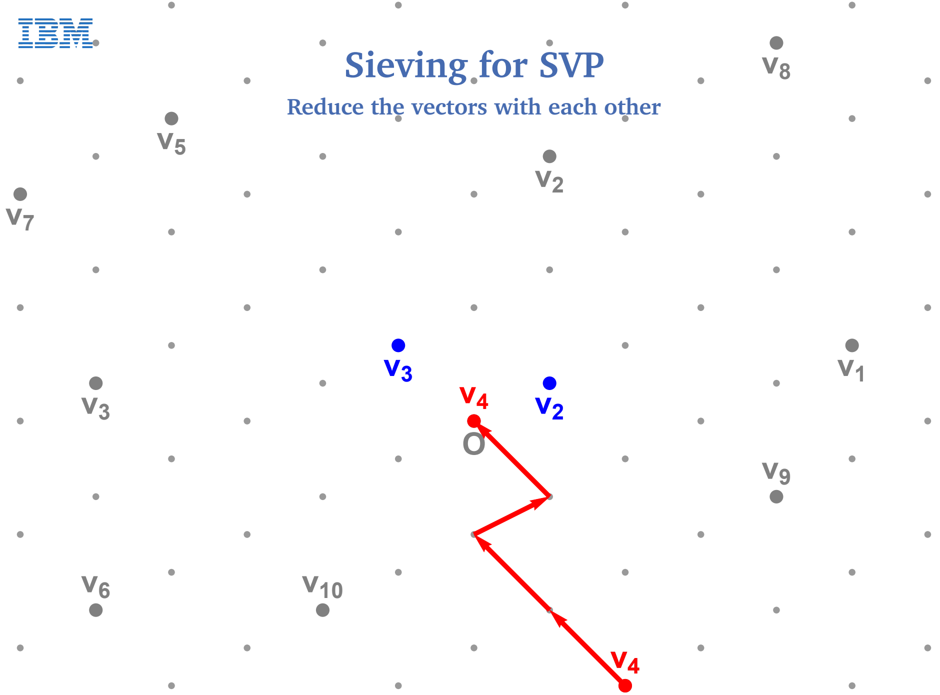
Sieving for SVP

Reduce the vectors with each other



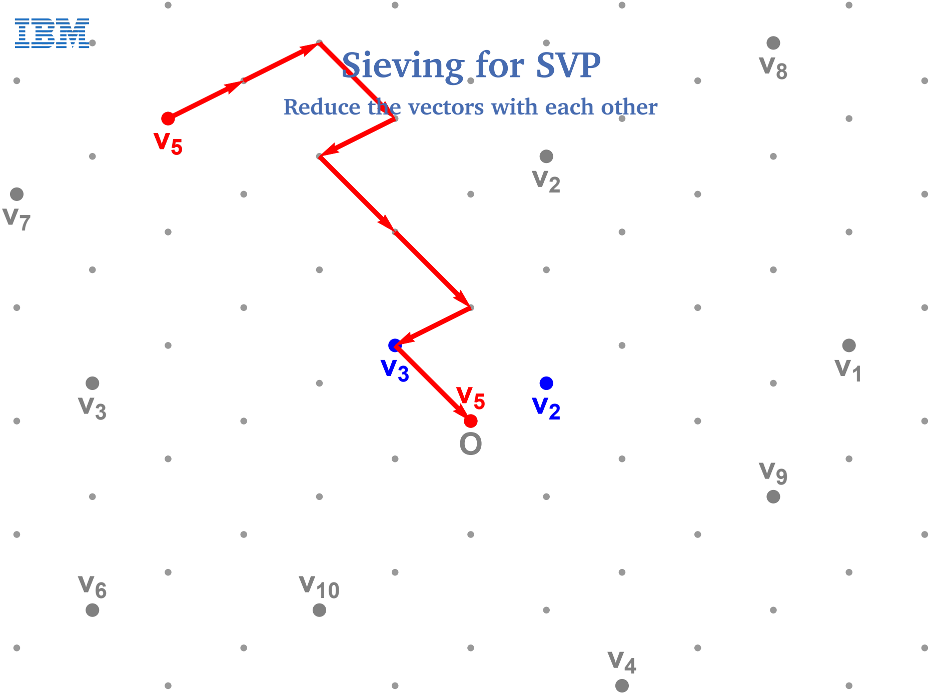
Sieving for SVP

Reduce the vectors with each other



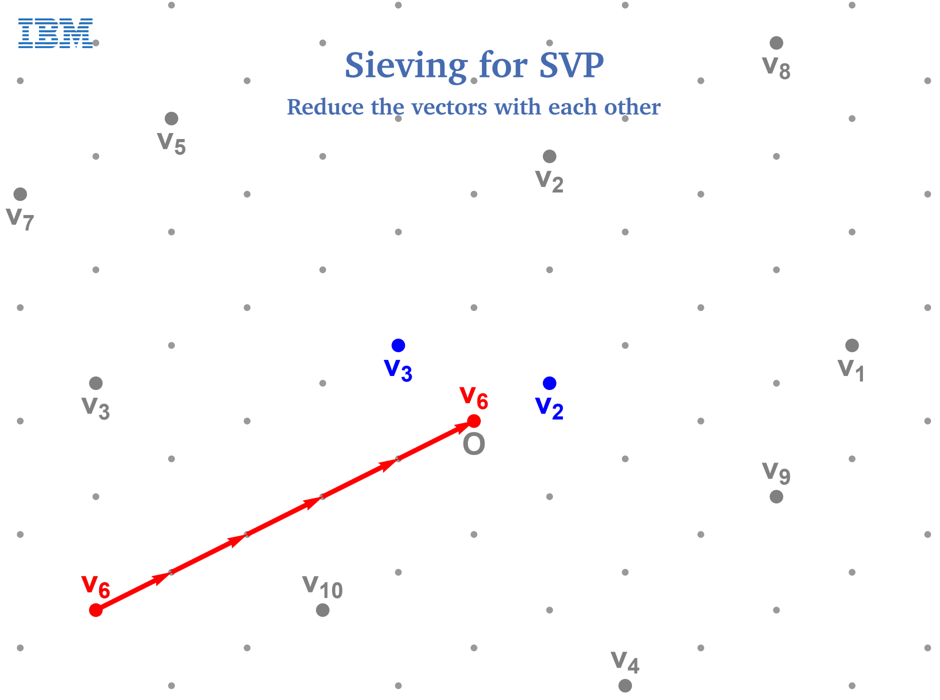
Sieving for SVP

Reduce the vectors with each other



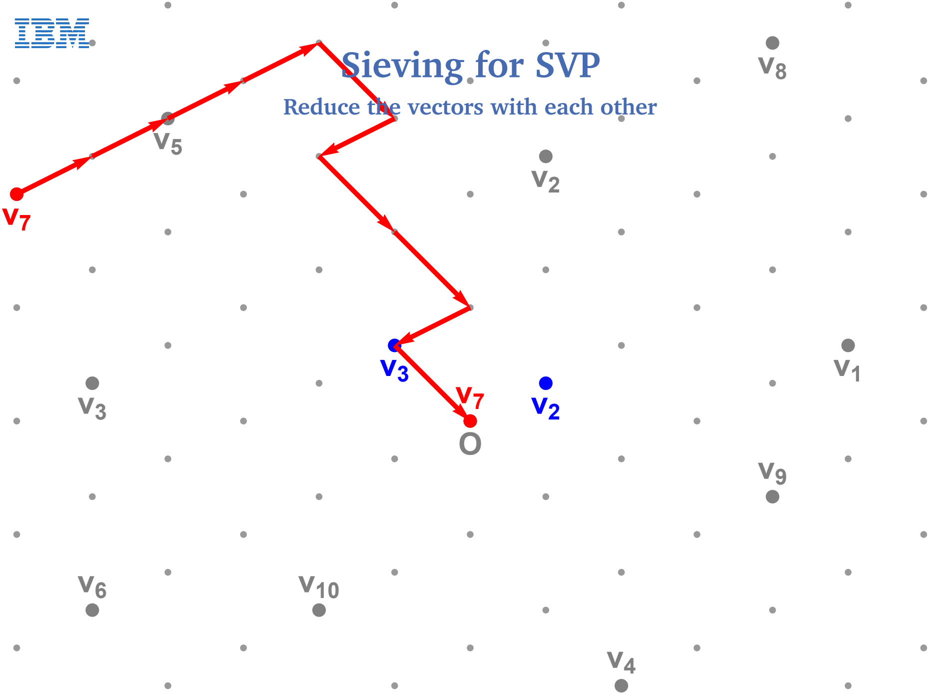
Sieving for SVP

Reduce the vectors with each other



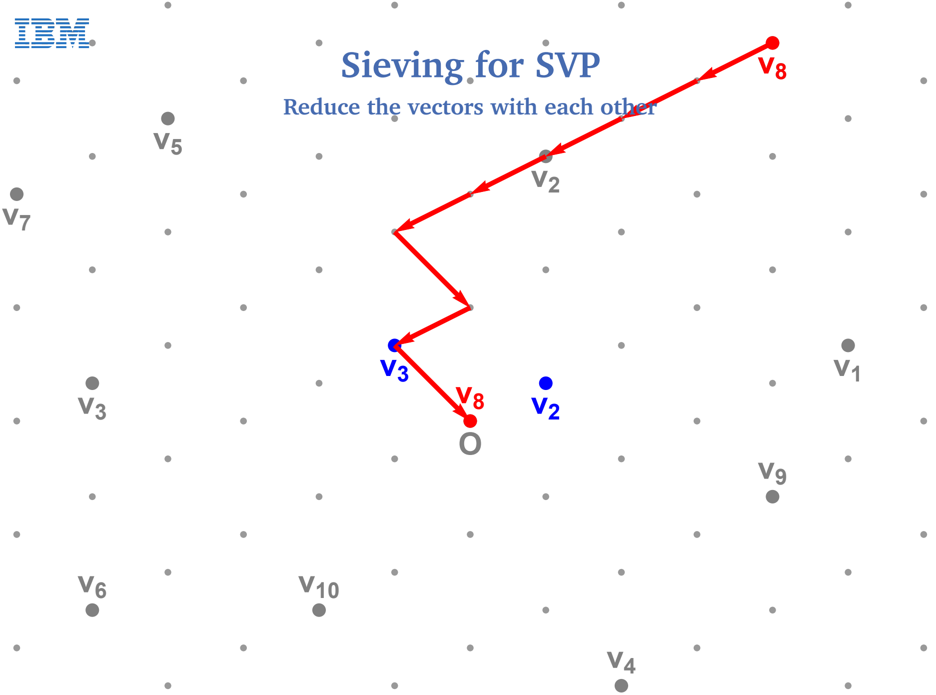
Sieving for SVP

Reduce the vectors with each other



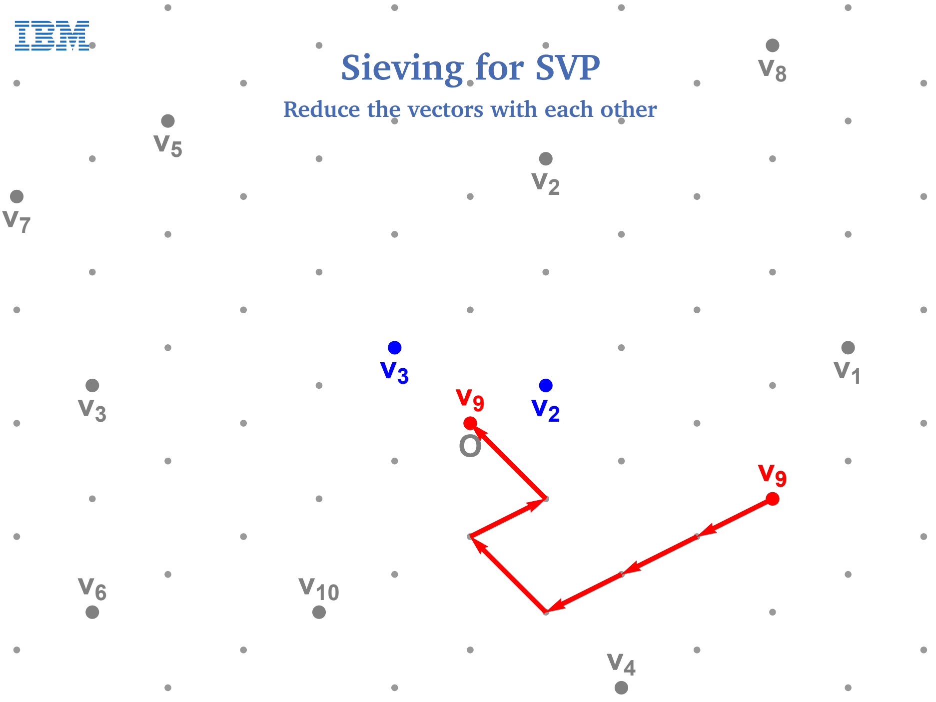
Sieving for SVP

Reduce the vectors with each other



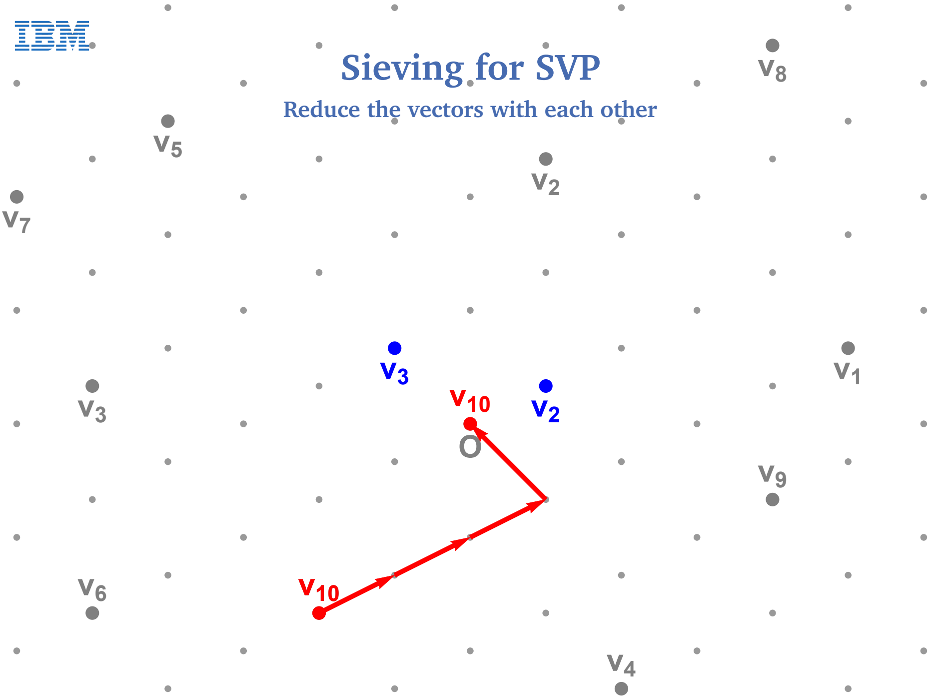
Sieving for SVP

Reduce the vectors with each other



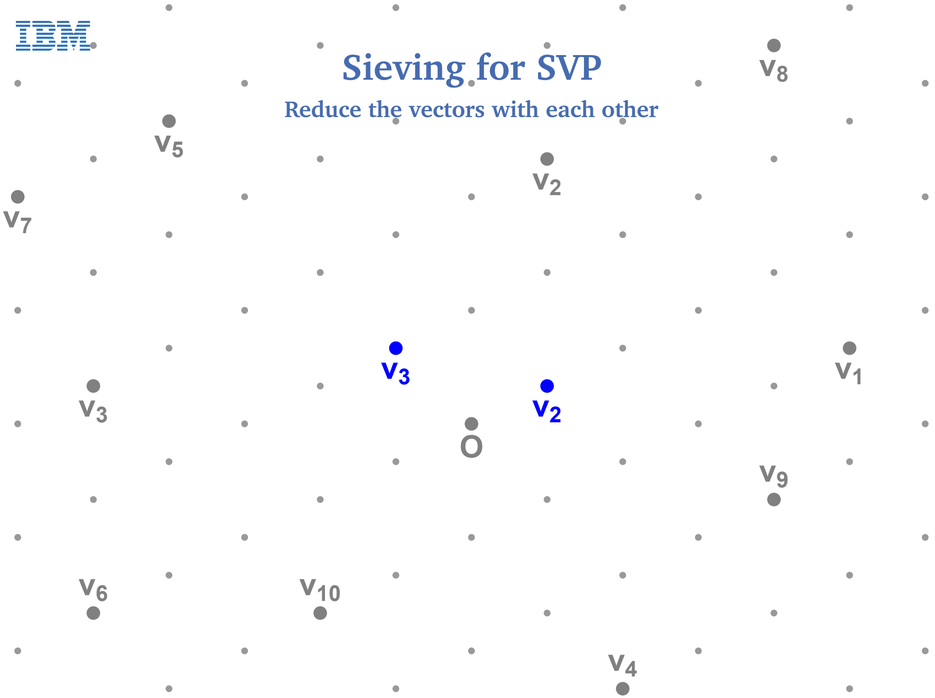
Sieving for SVP

Reduce the vectors with each other



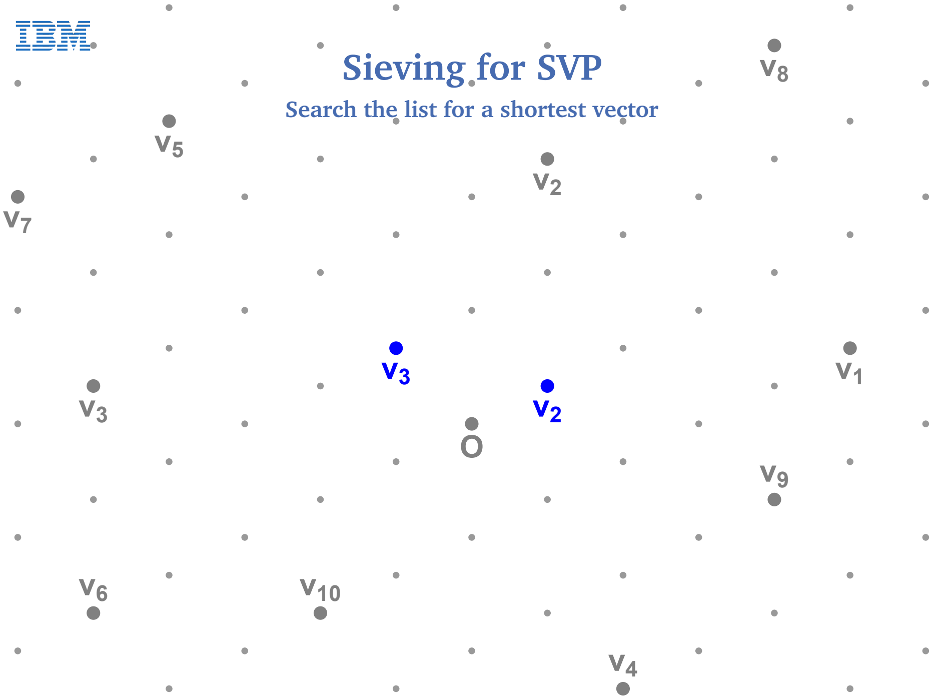
Sieving for SVP

Reduce the vectors with each other



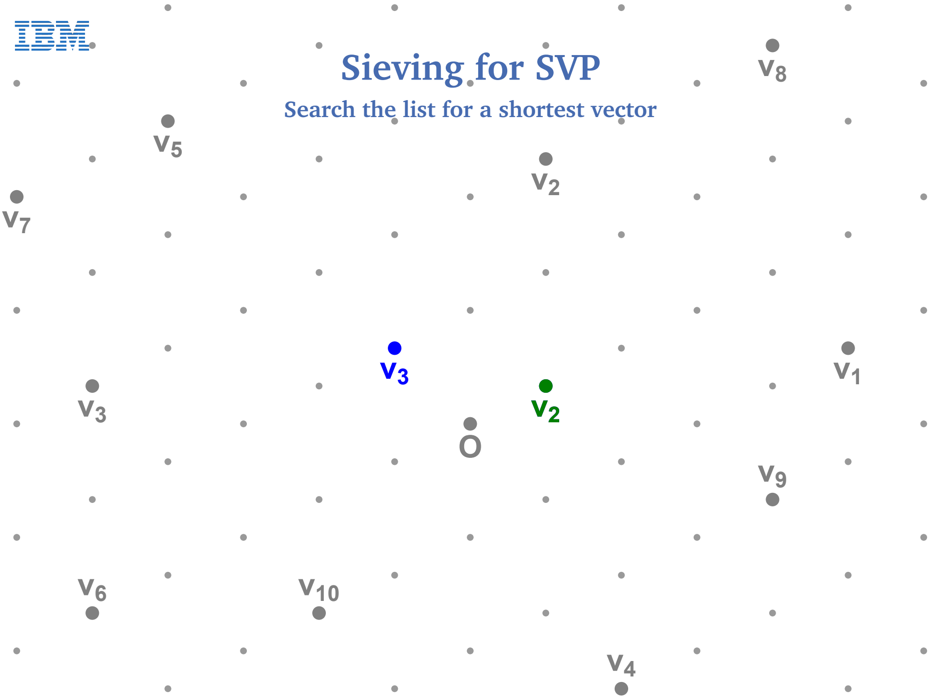
Sieving for SVP

Search the list for a shortest vector



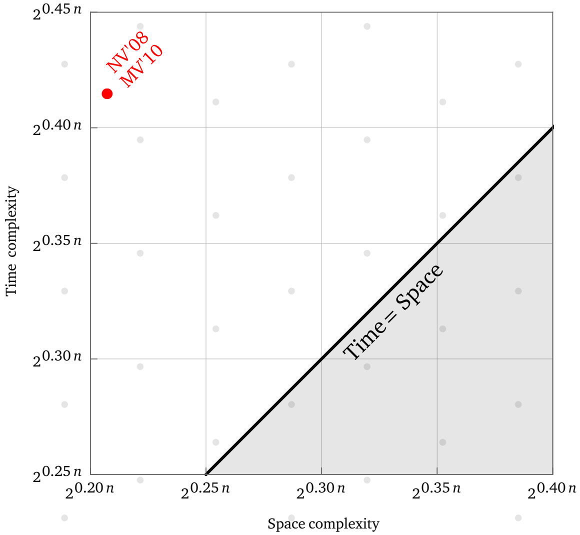
Sieving for SVP

Search the list for a shortest vector



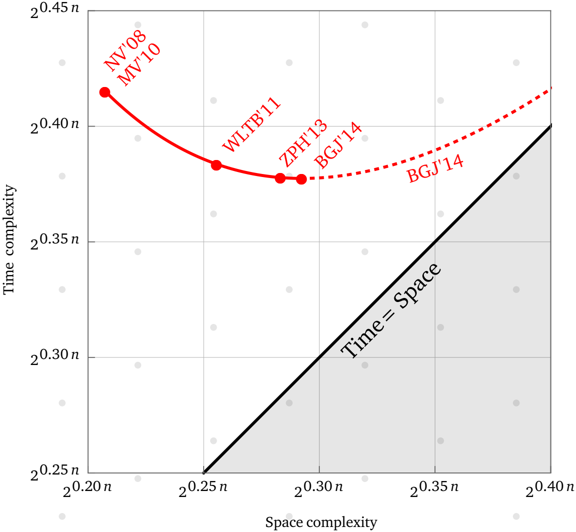
Sieving for SVP

The GaussSieve and Nguyen-Vidick sieve



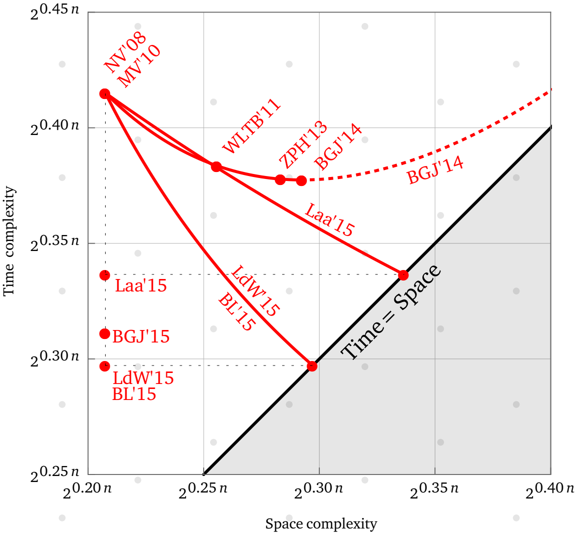
Sieving for SVP

Leveled sieving approaches



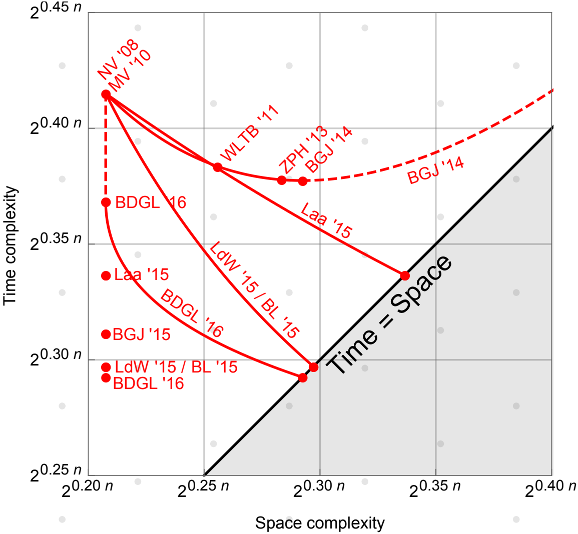
Sieving for SVP

Locality-Sensitive Hashing (LSH)



Sieving for SVP

Locality-Sensitive Filters (LSF)



Outline

Sieving for SVP

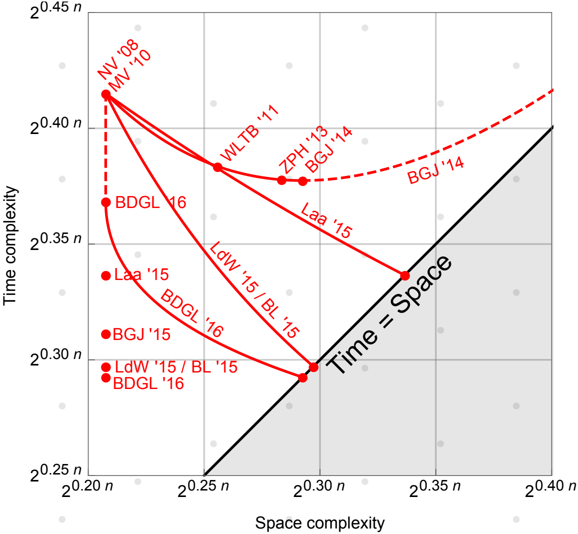
Sieving for CVP

Sieving for CVPP

Conclusion

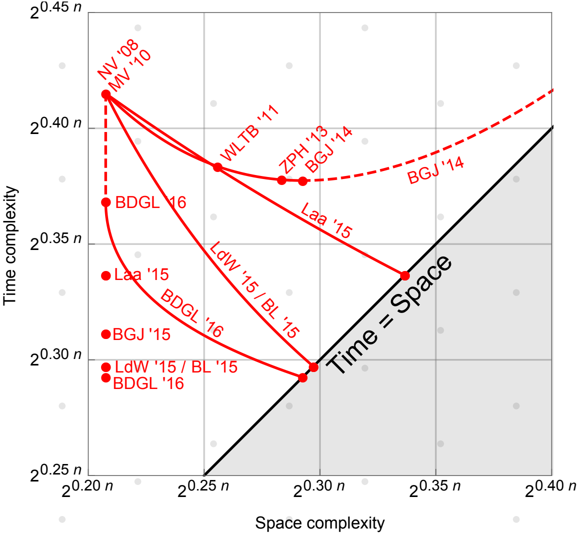
Sieving for SVP

Space/time trade-offs



Sieving for CVP

Space/time trade-offs



Sieving for CVP

- Intuitively, $\mathrm{CVP}_n \approx \mathrm{SVP}_{n+1}$ [Kan87]

Sieving for CVP

- Intuitively, $\mathrm{CVP}_n \approx \mathrm{SVP}_{n+1}$ [Kan87]
- Can also directly modify sieving to solve CVP

Sieving for CVP

- Intuitively, $\mathrm{CVP}_n \approx \mathrm{SVP}_{n+1}$ [Kan87]
- Can also directly modify sieving to solve CVP
- Costs of CVP_n factor 2 more than SVP_n

Outline

Sieving for SVP

Sieving for CVP

Sieving for CVPP

Conclusion

Sieving for CVPP

Run a GaussSieve as preprocessing

Sieving for CVPP

Run a GaussSieve as preprocessing

Sieving for CVPP

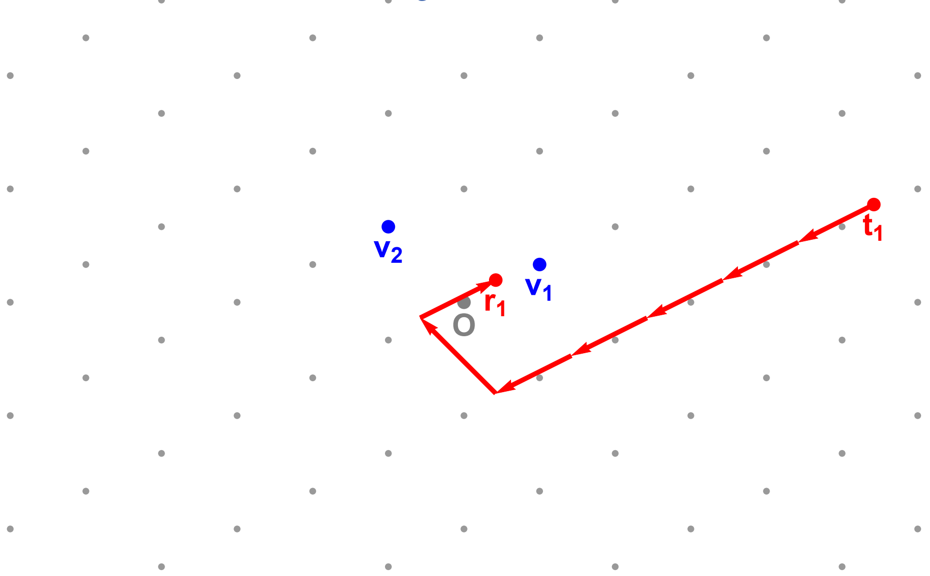
Reduce the target vectors with the list

Sieving for CVPP

Reduce the target vectors with the list

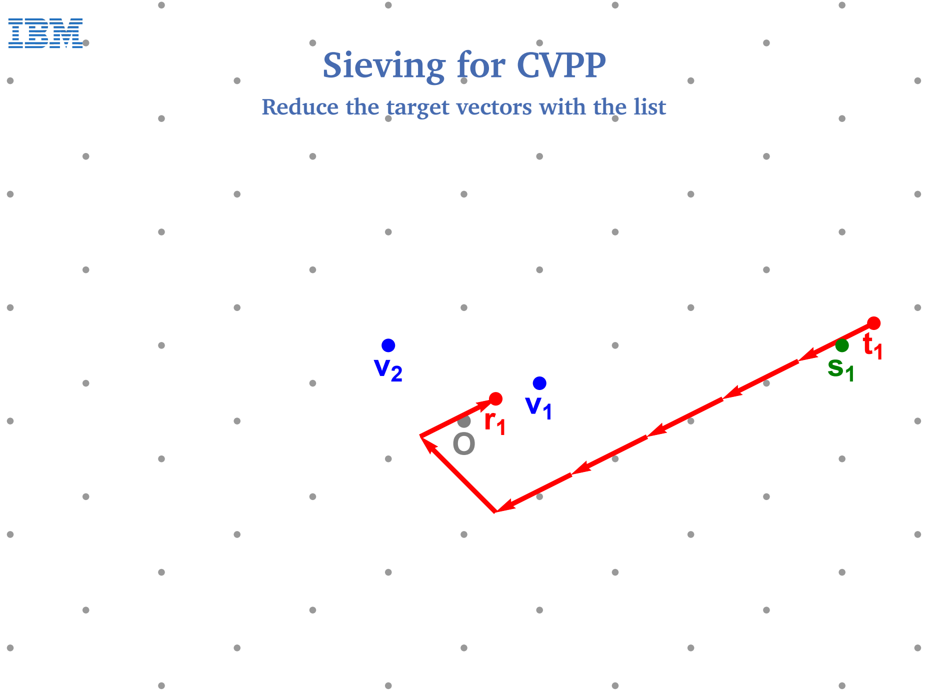
Sieving for CVPP

Reduce the target vectors with the list



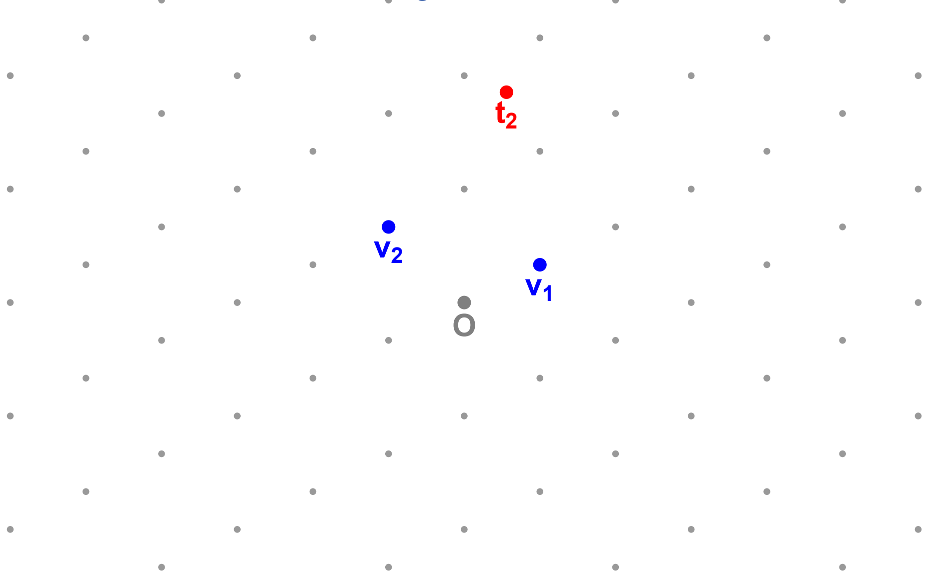
Sieving for CVPP

Reduce the target vectors with the list



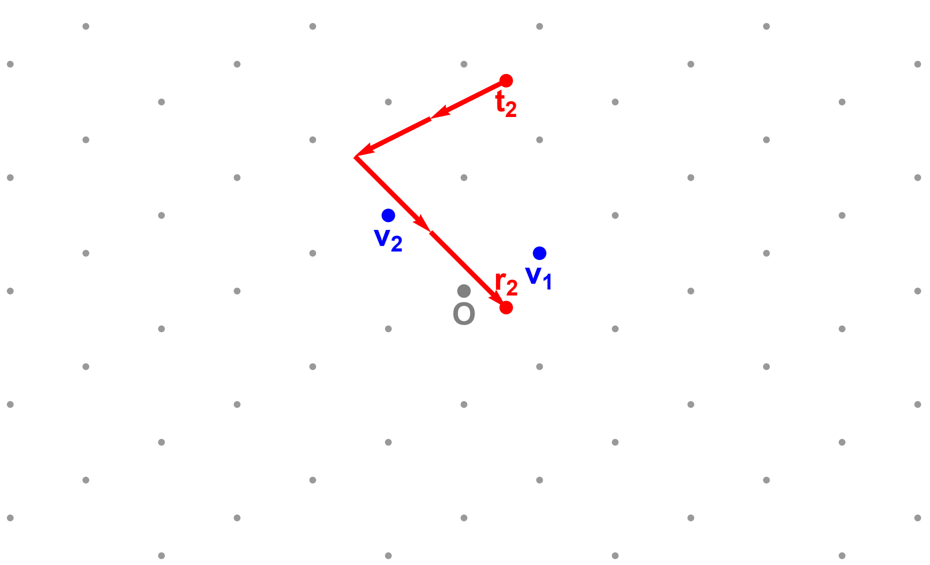
Sieving for CVPP

Reduce the target vectors with the list



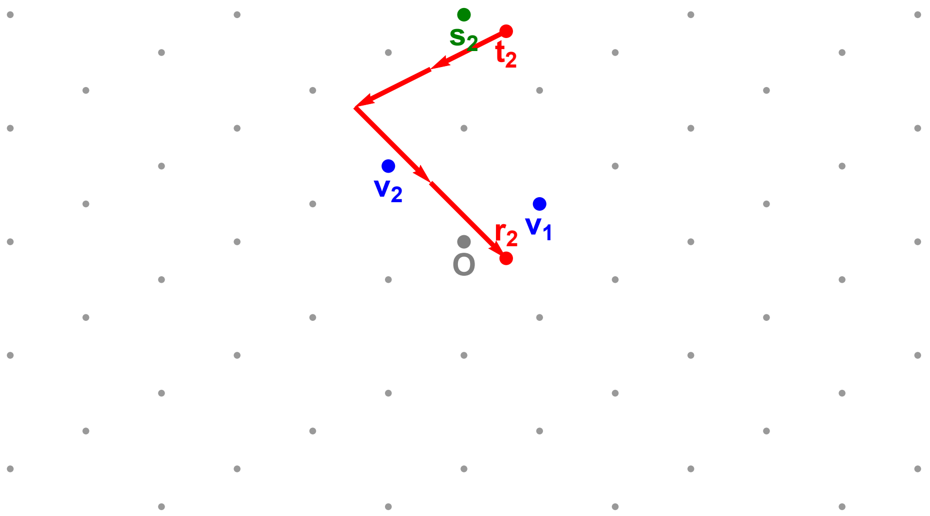
Sieving for CVPP

Reduce the target vectors with the list



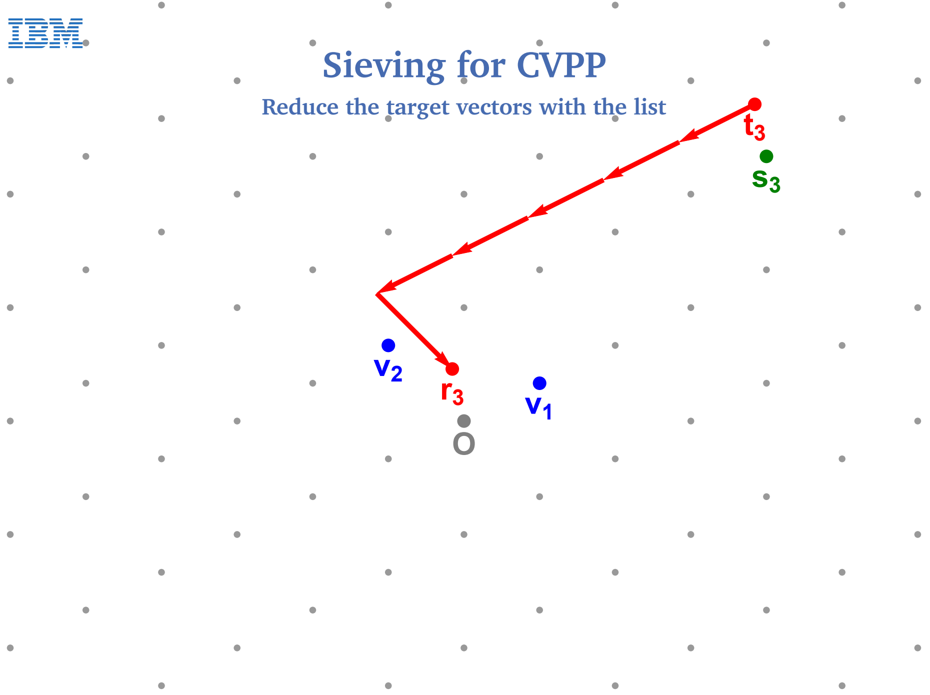
Sieving for CVPP

Reduce the target vectors with the list



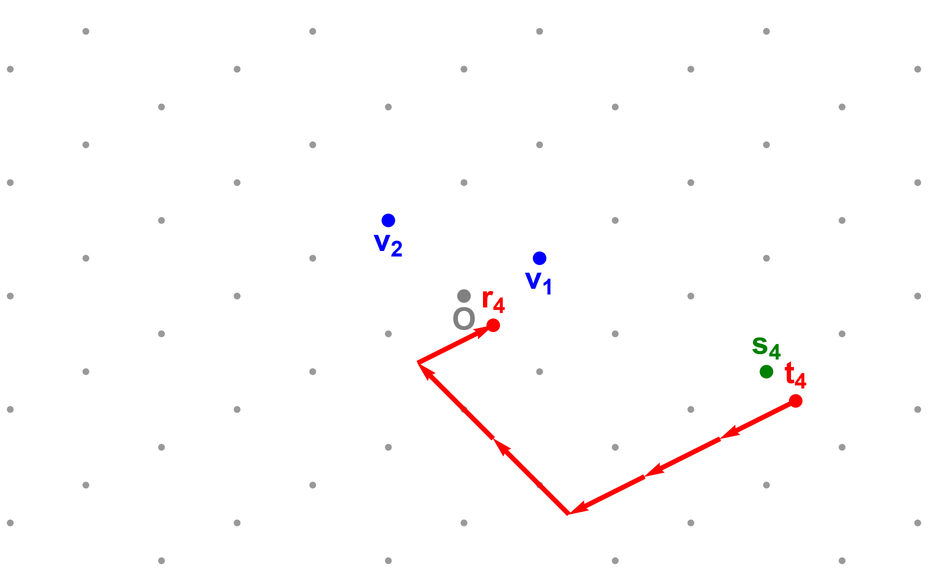
Sieving for CVPP

Reduce the target vectors with the list



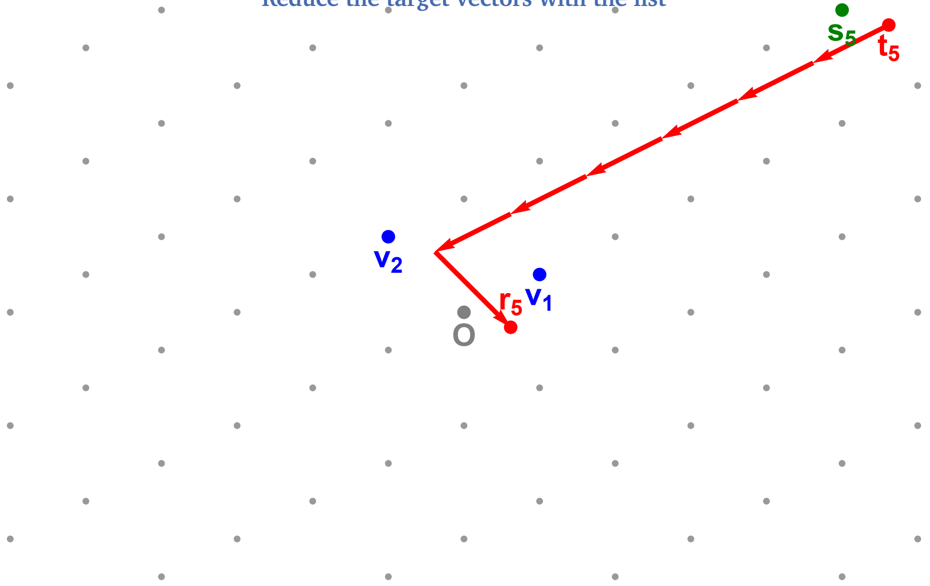
Sieving for CVPP

Reduce the target vectors with the list



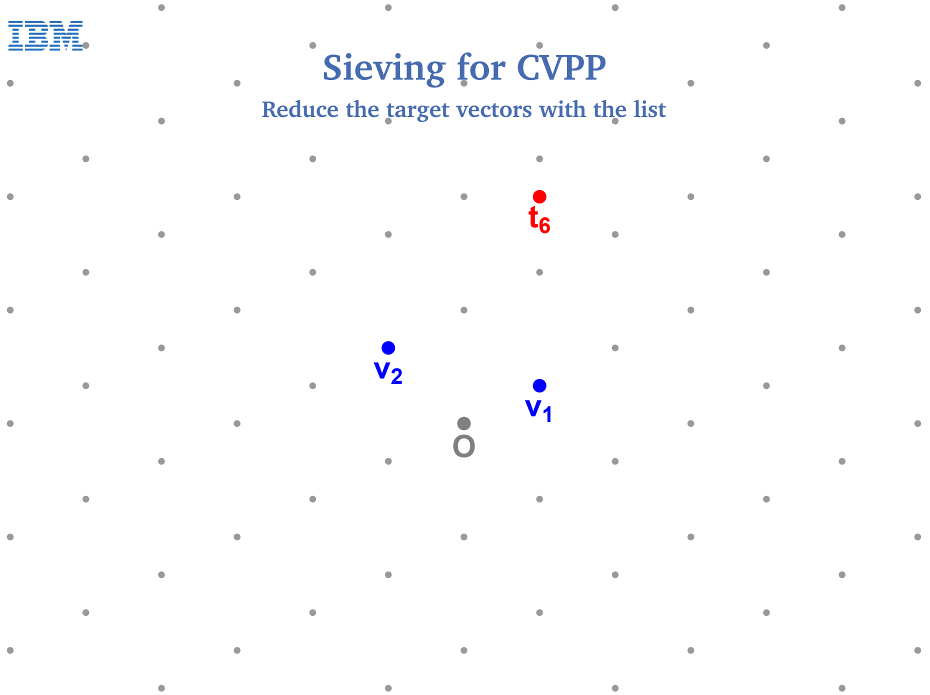
Sieving for CVPP

Reduce the target vectors with the list



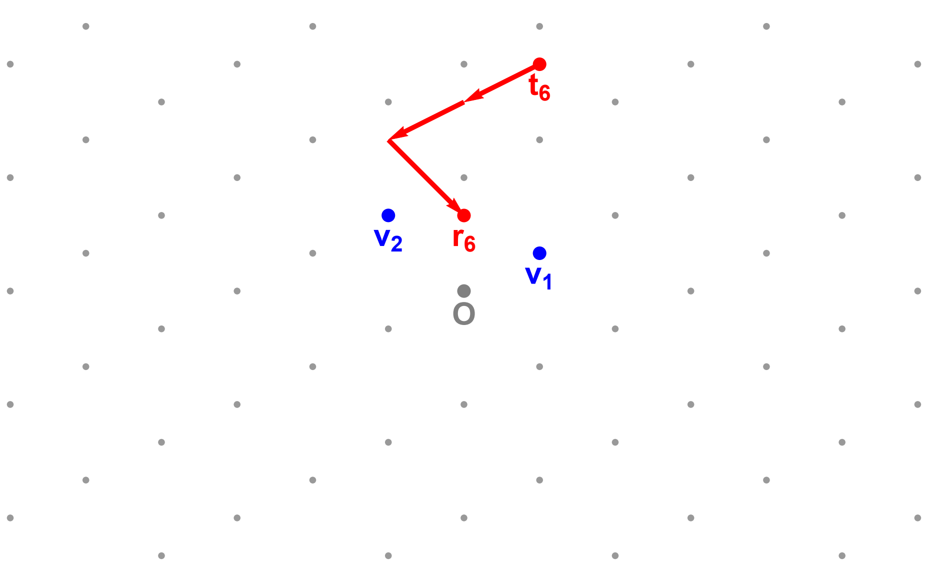
Sieving for CVPP

Reduce the target vectors with the list



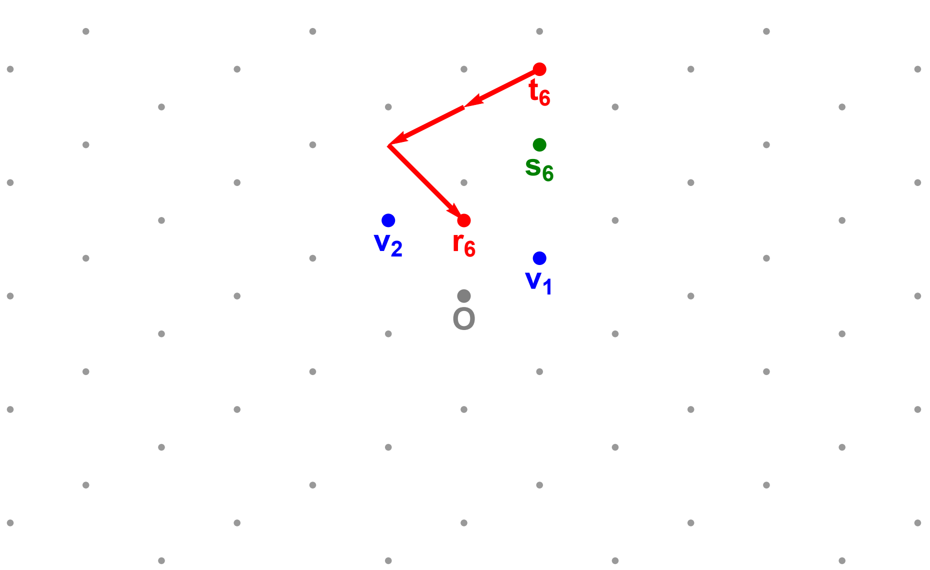
Sieving for CVPP

Reduce the target vectors with the list



Sieving for CVPP

Reduce the target vectors with the list

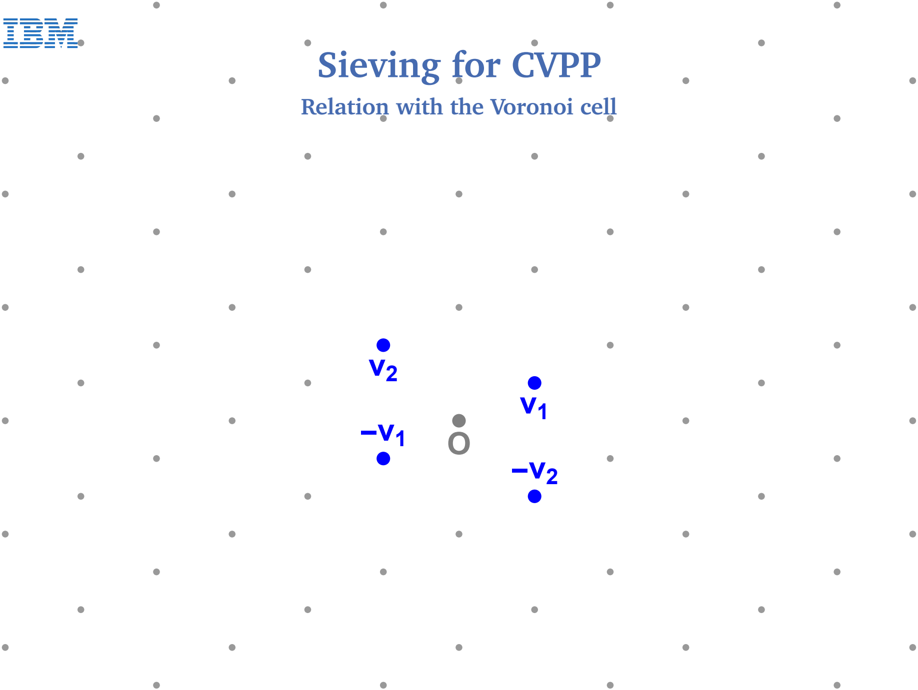


Sieving for CVPP

Relation with the Voronoi cell

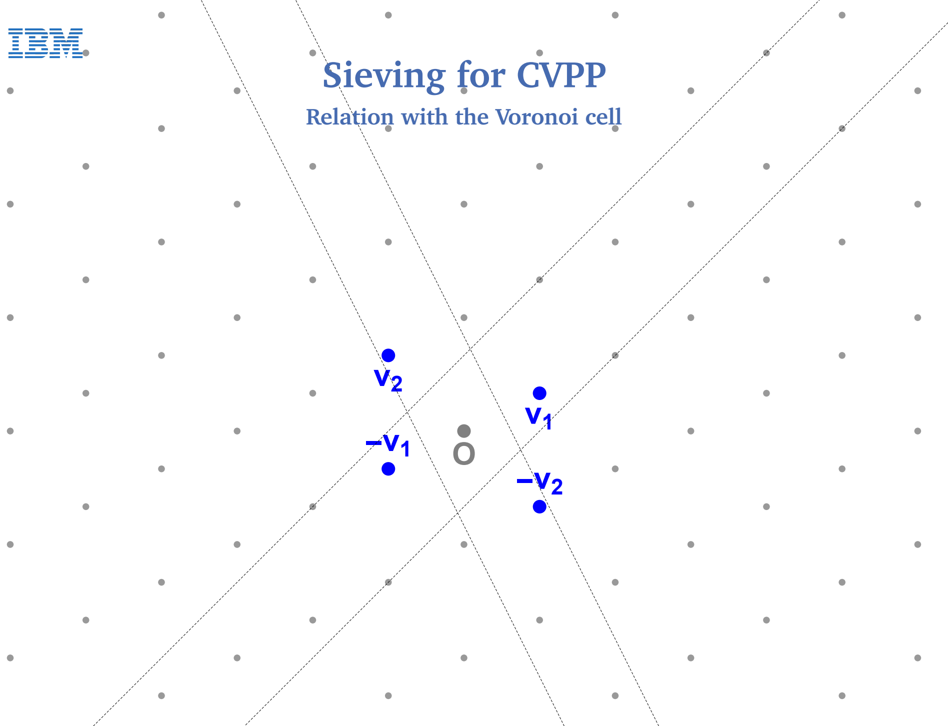
Sieving for CVPP

Relation with the Voronoi cell



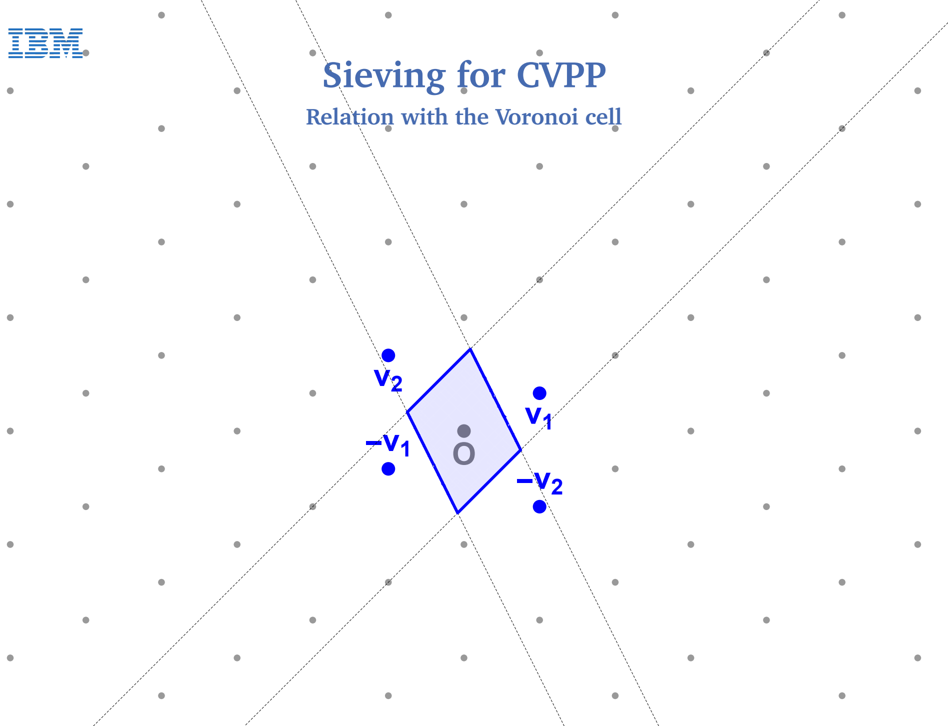
Sieving for CVPP

Relation with the Voronoi cell



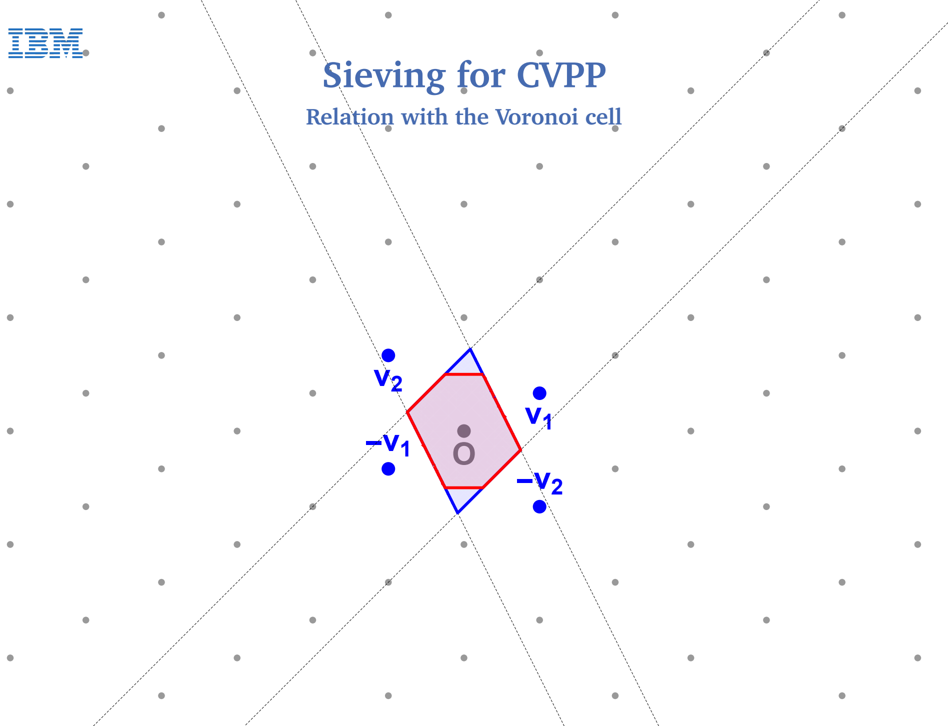
Sieving for CVPP

Relation with the Voronoi cell



Sieving for CVPP

Relation with the Voronoi cell



Sieving for CVPP

Overview

Sieving for CVPP

Overview

- Blue region: **Gauss cell** $\mathscr{G}$

Sieving for CVPP

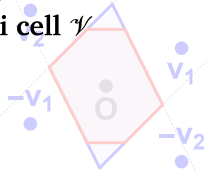
Overview

- Blue region: **Gauss cell** $\mathscr{G}$
  - Defined by $2^{0.21n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{G}) = 2^{O(n)} \cdot \det(\mathscr{L})$
  - Reductions always land in $\mathscr{G}$

Sieving for CVPP

Overview

- Blue region: **Gauss cell** $\mathscr{G}$
  - Defined by $2^{0.21n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{G}) = 2^{O(n)} \cdot \det(\mathscr{L})$
  - Reductions always land in $\mathscr{G}$
- Red region: **Voronoi cell** $\mathscr{V}$



Sieving for CVPP

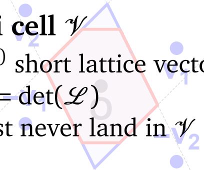
Overview

- Blue region: **Gauss cell** $\mathscr{G}$
  - Defined by $2^{0.21n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{G}) = 2^{O(n)} \cdot \det(\mathscr{L})$
  - Reductions always land in $\mathscr{G}$
- Red region: **Voronoi cell** $\mathscr{V}$
  - Defined by $2^{n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{V}) = \det(\mathscr{L})$
  - Reductions almost never land in $\mathscr{V}$

Sieving for CVPP

Overview

- Blue region: **Gauss cell** $\mathscr{G}$
  - Defined by $2^{0.21n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{G}) = 2^{O(n)} \cdot \det(\mathscr{L})$
  - Reductions always land in $\mathscr{G}$
- Red region: **Voronoi cell** $\mathscr{V}$
  - Defined by $2^{n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{V}) = \det(\mathscr{L})$
  - Reductions almost never land in $\mathscr{V}$
- Problems:



Sieving for CVPP

Overview

- Blue region: **Gauss cell** $\mathscr{G}$
  - Defined by $2^{0.21n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{G}) = 2^{O(n)} \cdot \det(\mathscr{L})$
  - Reductions always land in $\mathscr{G}$
- Red region: **Voronoi cell** $\mathscr{V}$
  - Defined by $2^{n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{V}) = \det(\mathscr{L})$
  - Reductions almost never land in $\mathscr{V}$
- Problems:
  - Exponentially small success probability $\mathrm{Vol}(\mathscr{V})/\mathrm{Vol}(\mathscr{G})$

Sieving for CVPP

Overview

- Blue region: **Gauss cell** $\mathscr{G}$
  - Defined by $2^{0.21n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{G}) = 2^{O(n)} \cdot \det(\mathscr{L})$
  - Reductions always land in $\mathscr{G}$
- Red region: **Voronoi cell** $\mathscr{V}$
  - Defined by $2^{n+o(n)}$ short lattice vectors
  - Volume: $\mathrm{Vol}(\mathscr{V}) = \det(\mathscr{L})$
  - Reductions almost never land in $\mathscr{V}$
- Problems:
  - Exponentially small success probability $\mathrm{Vol}(\mathscr{V})/\mathrm{Vol}(\mathscr{G})$
  - Probability only over randomness of targets

Sieving for CVPP

Solving the problems

- Idea 1: **Larger lists, weaker reductions**

Sieving for CVPP

Solving the problems

- Idea 1: **Larger lists, weaker reductions**
  - Problem: Exponentially small success probability

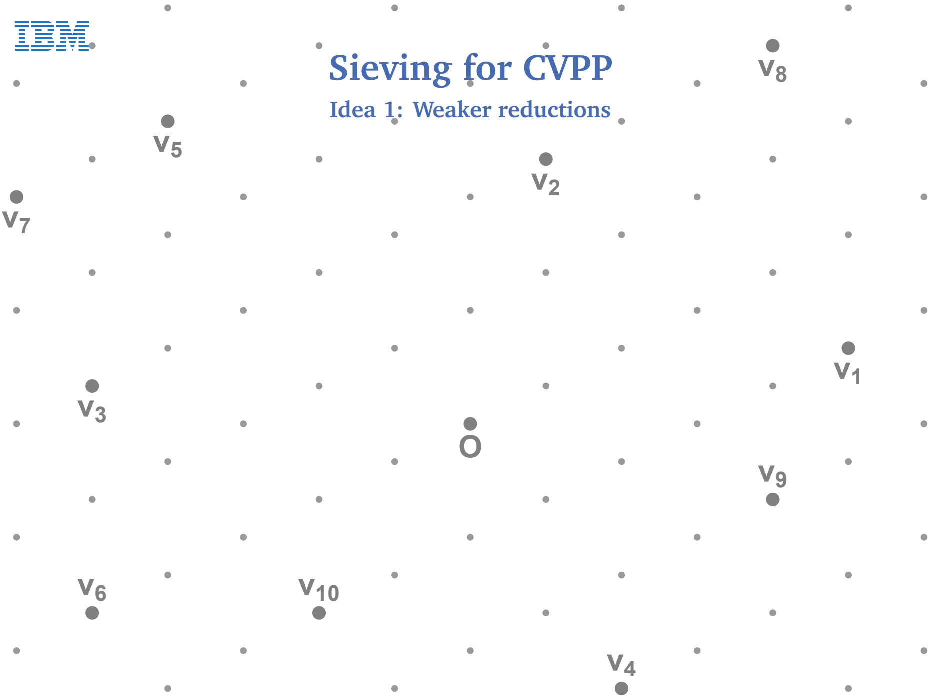
Sieving for CVPP

Solving the problems

- Idea 1: **Larger lists, weaker reductions**
  - Problem: Exponentially small success probability
  - To guarantee $\mathrm{Vol}(\mathscr{G}) \approx \mathrm{Vol}(\mathscr{V})$, need $2^{n/2+o(n)}$ vectors
  - Preprocessing: reduce $\boldsymbol{v}_1$ with $\boldsymbol{v}_2$ iff $\|\boldsymbol{v}_1 - \boldsymbol{v}_2\| \leq (\sqrt{2-\sqrt{2}})\|\boldsymbol{v}_1\|$
  - Fewer reductions $\Longrightarrow$ NNS techniques work even better!

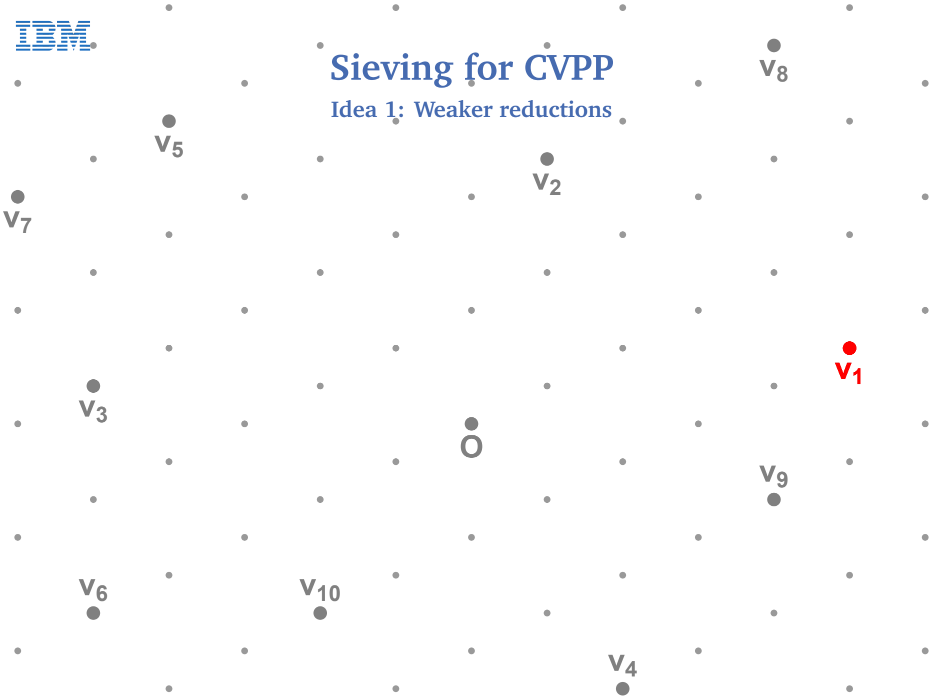
Sieving for CVPP

Idea 1: Weaker reductions



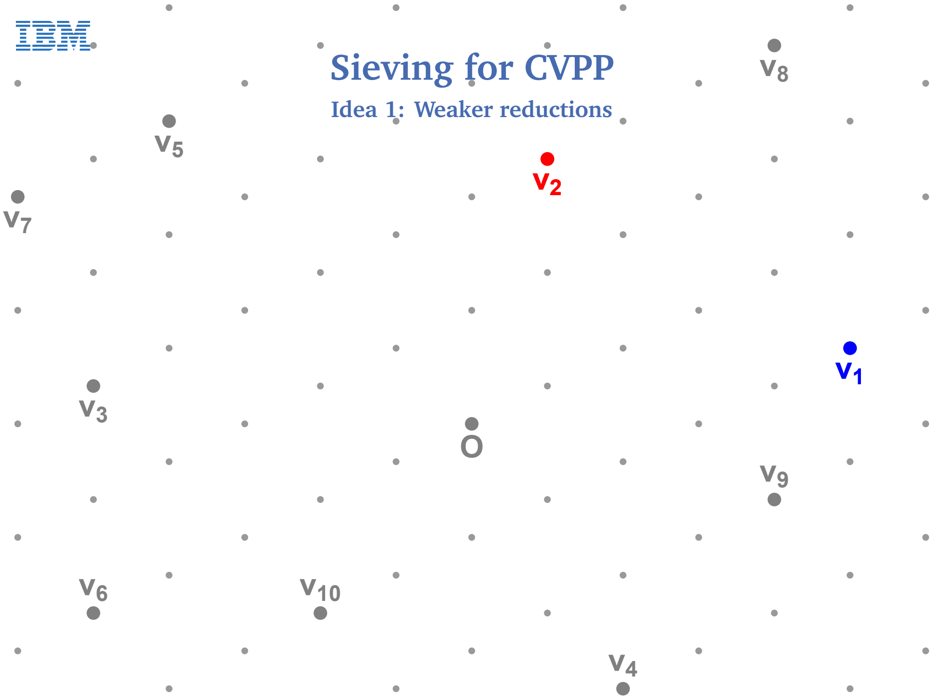
Sieving for CVPP

Idea 1: Weaker reductions



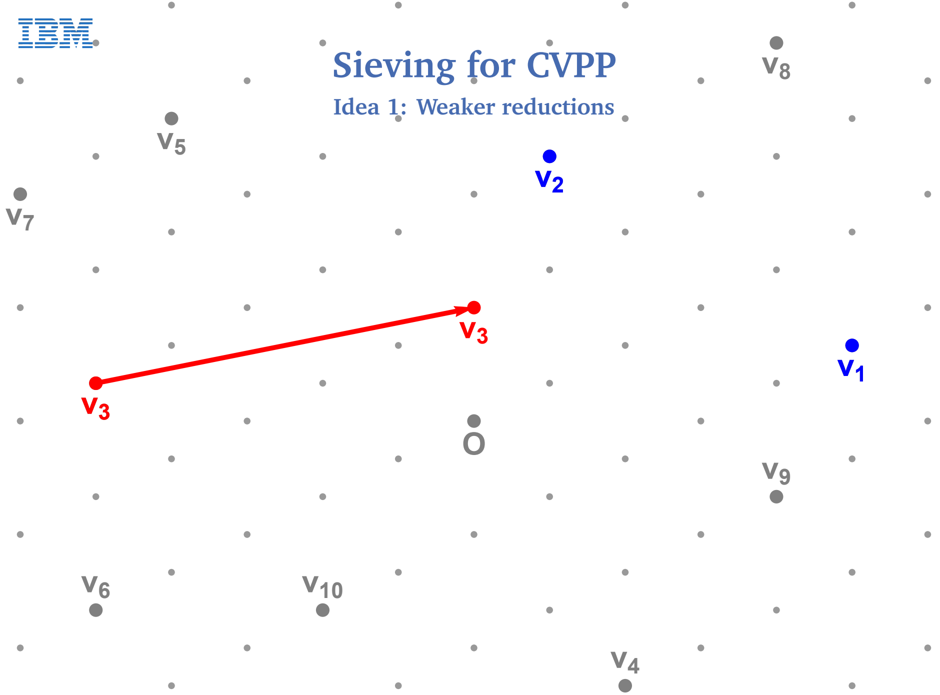
Sieving for CVPP

Idea 1: Weaker reductions



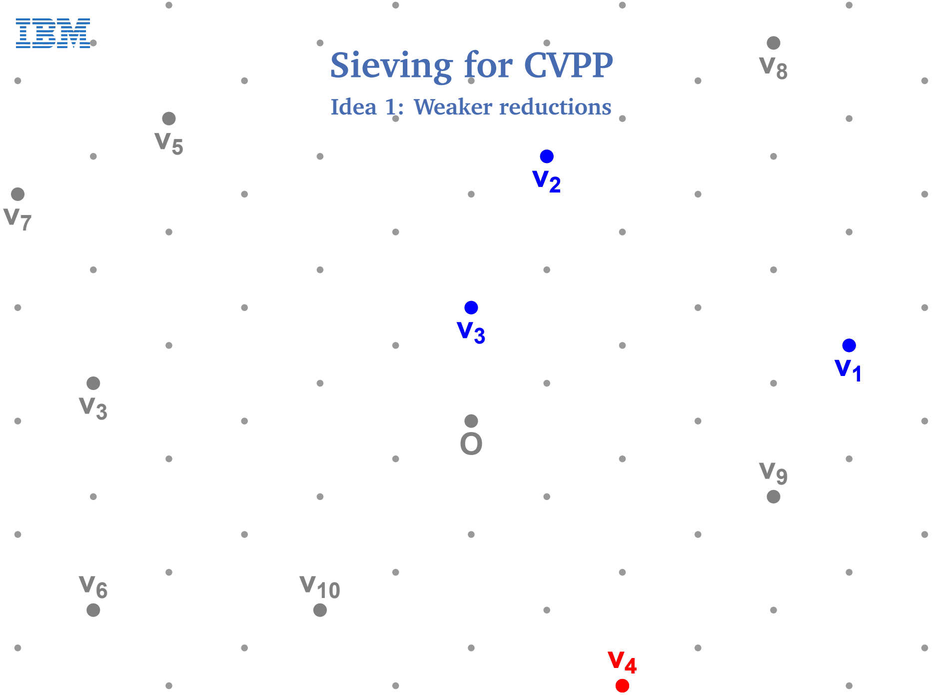
Sieving for CVPP

Idea 1: Weaker reductions



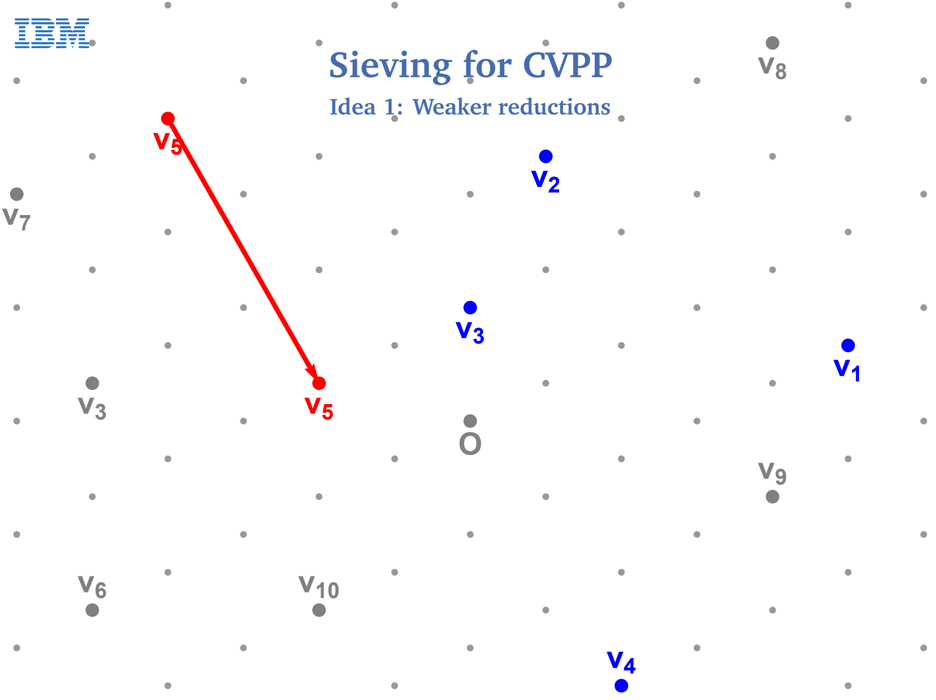
Sieving for CVPP

Idea 1: Weaker reductions



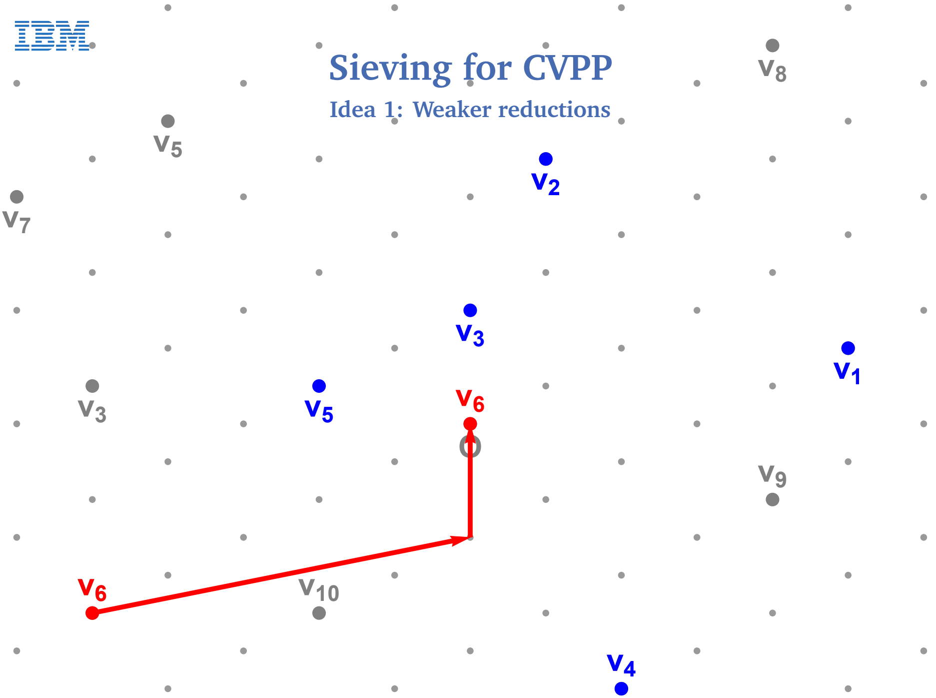
Sieving for CVPP

Idea 1: Weaker reductions



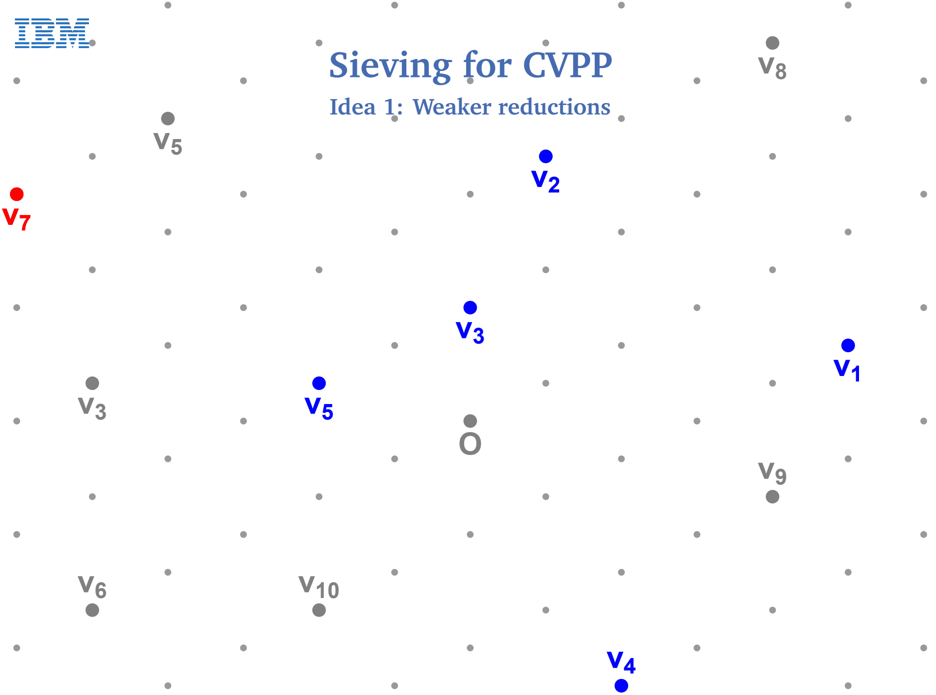
Sieving for CVPP

Idea 1: Weaker reductions



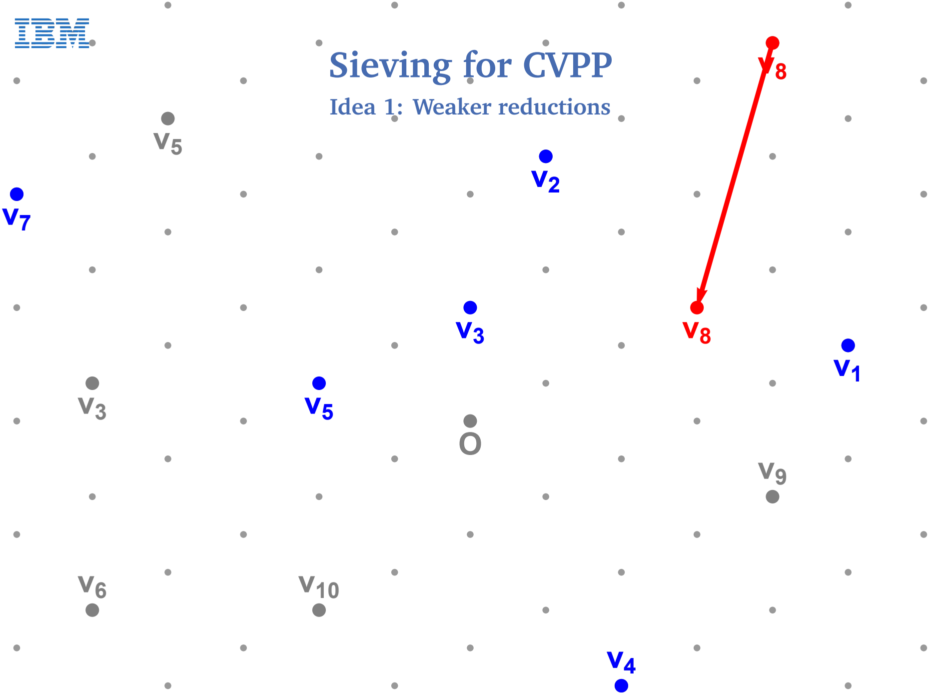
Sieving for CVPP

Idea 1: Weaker reductions



Sieving for CVPP

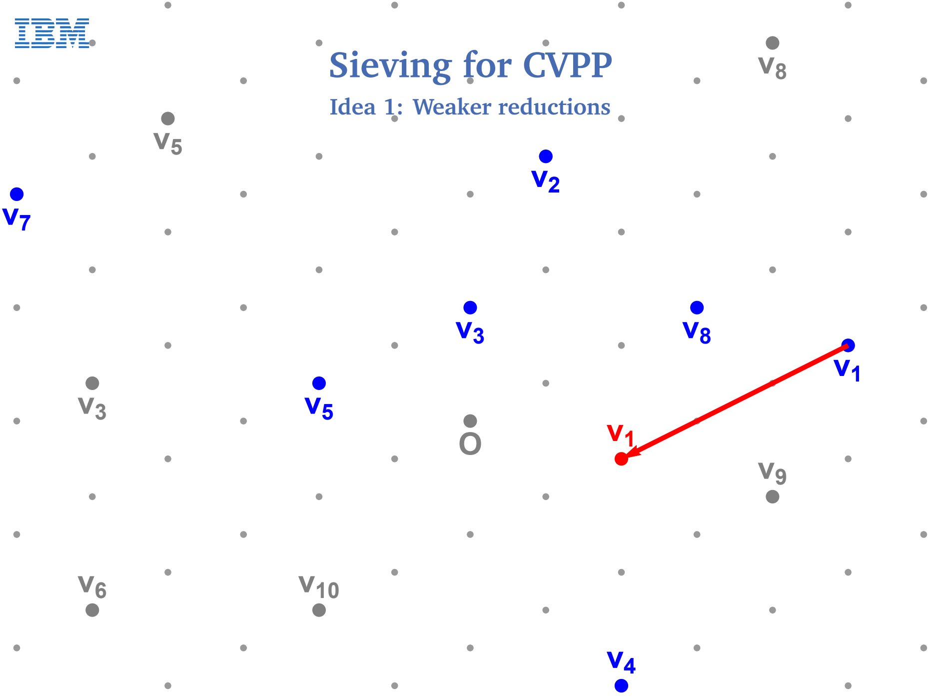
Idea 1: Weaker reductions



IBM

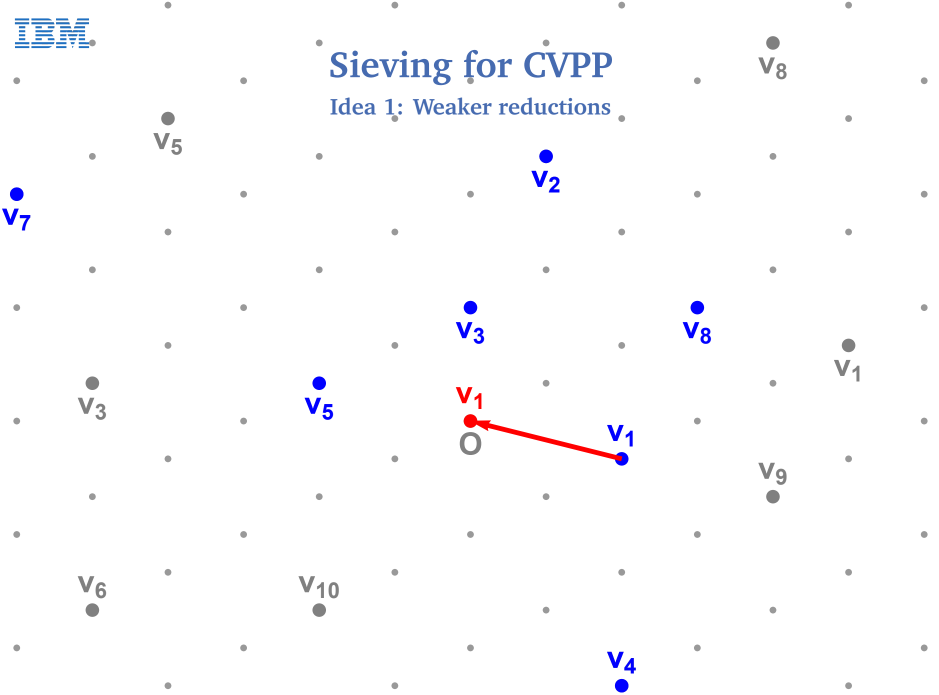
Sieving for CVPP

Idea 1: Weaker reductions



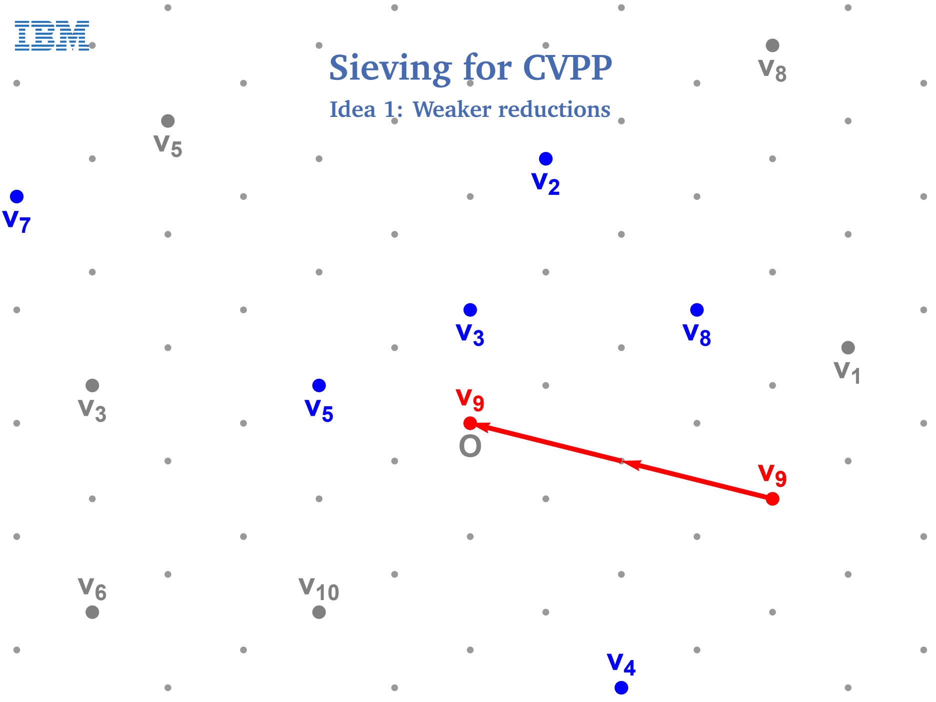
Sieving for CVPP

Idea 1: Weaker reductions



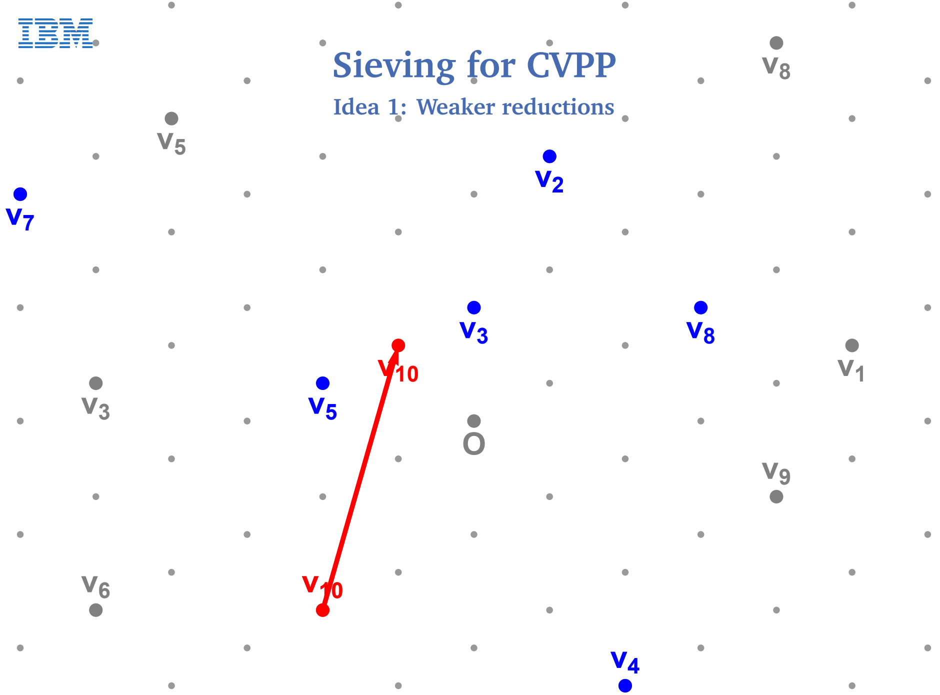
Sieving for CVPP

Idea 1: Weaker reductions



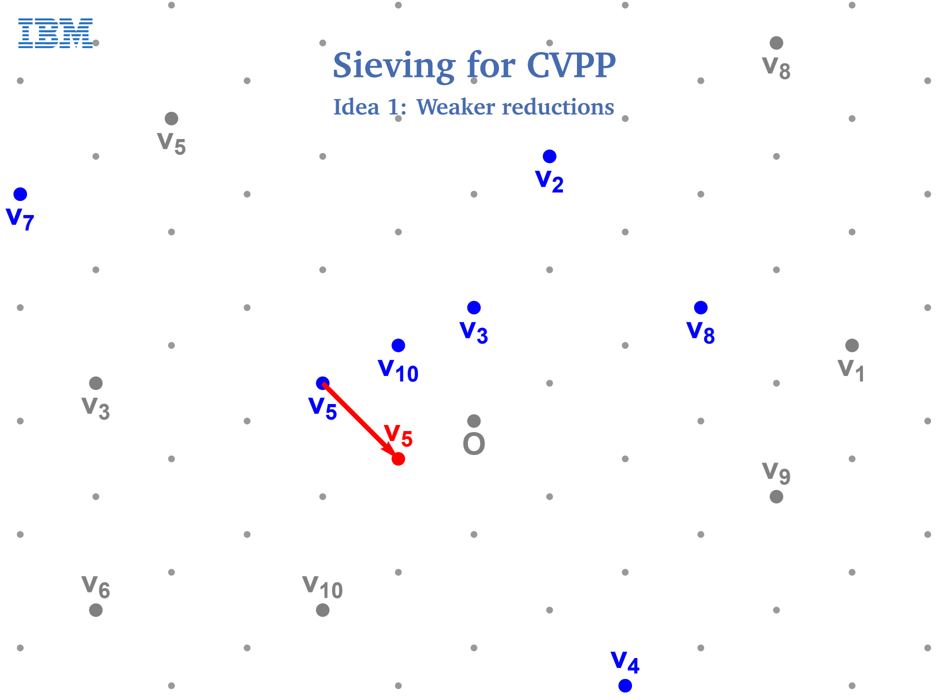
Sieving for CVPP

Idea 1: Weaker reductions



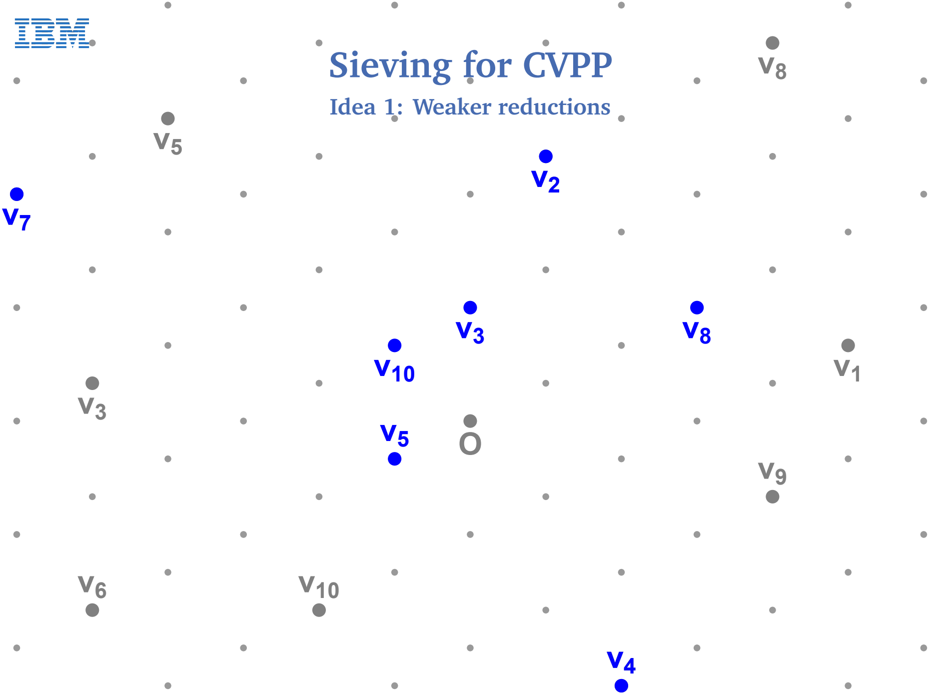
Sieving for CVPP

Idea 1: Weaker reductions



Sieving for CVPP

Idea 1: Weaker reductions



Sieving for CVPP

Solving the problems

- Idea 1: **Larger lists, weaker reductions**
  - Problem: Exponentially small success probability
  - To guarantee $\mathrm{Vol}(\mathscr{G}) \approx \mathrm{Vol}(\mathscr{V})$, need $2^{n/2+o(n)}$ vectors
  - Preprocessing: reduce $\boldsymbol{v}_1$ with $\boldsymbol{v}_2$ iff $\|\boldsymbol{v}_1 - \boldsymbol{v}_2\| \leq (\sqrt{2-\sqrt{2}})\|\boldsymbol{v}_1\|$
  - Fewer reductions $\implies$ NNS techniques work even better!

Sieving for CVPP

Solving the problems

- Idea 1: **Larger lists, weaker reductions**
  - Problem: Exponentially small success probability
  - To guarantee $\mathrm{Vol}(\mathscr{G}) \approx \mathrm{Vol}(\mathscr{V})$, need $2^{n/2+o(n)}$ vectors
  - Preprocessing: reduce $\boldsymbol{v}_1$ with $\boldsymbol{v}_2$ iff $\|\boldsymbol{v}_1 - \boldsymbol{v}_2\| \leq (\sqrt{2-\sqrt{2}})\|\boldsymbol{v}_1\|$
  - Fewer reductions $\implies$ NNS techniques work even better!
- Idea 2: **Rerandomizations** (full version)

Sieving for CVPP

Solving the problems

- Idea 1: **Larger lists, weaker reductions**
  - Problem: Exponentially small success probability
  - To guarantee $\text{Vol}(\mathcal{G}) \approx \text{Vol}(\mathcal{V})$, need $2^{n/2+o(n)}$ vectors
  - Preprocessing: reduce $\boldsymbol{v}_1$ with $\boldsymbol{v}_2$ iff $\|\boldsymbol{v}_1 - \boldsymbol{v}_2\| \leq (\sqrt{2-\sqrt{2}})\|\boldsymbol{v}_1\|$
  - Fewer reductions $\Longrightarrow$ NNS techniques work even better!
- Idea 2: **Rerandomizations** (full version)
  - Problem: Probability only over randomness of targets

Sieving for CVPP

Solving the problems

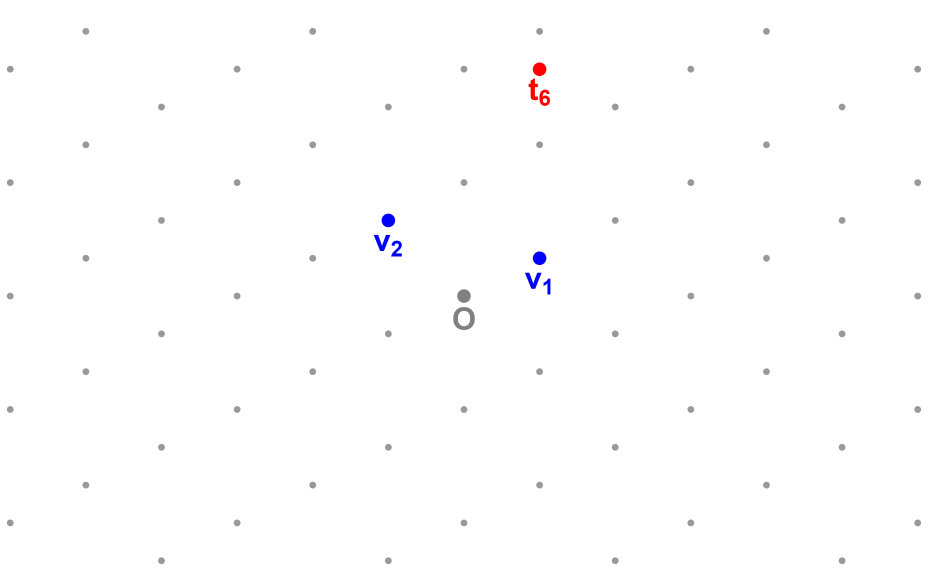
- Idea 1: **Larger lists, weaker reductions**
  - Problem: Exponentially small success probability
  - To guarantee $\mathrm{Vol}(\mathcal{G}) \approx \mathrm{Vol}(\mathcal{V})$, need $2^{n/2+o(n)}$ vectors
  - Preprocessing: reduce $\boldsymbol{v}_1$ with $\boldsymbol{v}_2$ iff $\|\boldsymbol{v}_1 - \boldsymbol{v}_2\| \leq (\sqrt{2-\sqrt{2}})\|\boldsymbol{v}_1\|$
  - Fewer reductions $\Longrightarrow$ NNS techniques work even better!
- Idea 2: **Rerandomizations** (full version)
  - Problem: Probability only over randomness of targets
  - Randomize target $\boldsymbol{t}$ before reducing ($\boldsymbol{t}' \in_R \boldsymbol{t} + \mathcal{L}$)
  - Randomness now over algorithm, independently of target
  - Optimize expected time (time / success probability)

Sieving for CVPP

Idea 2: Rerandomize the target

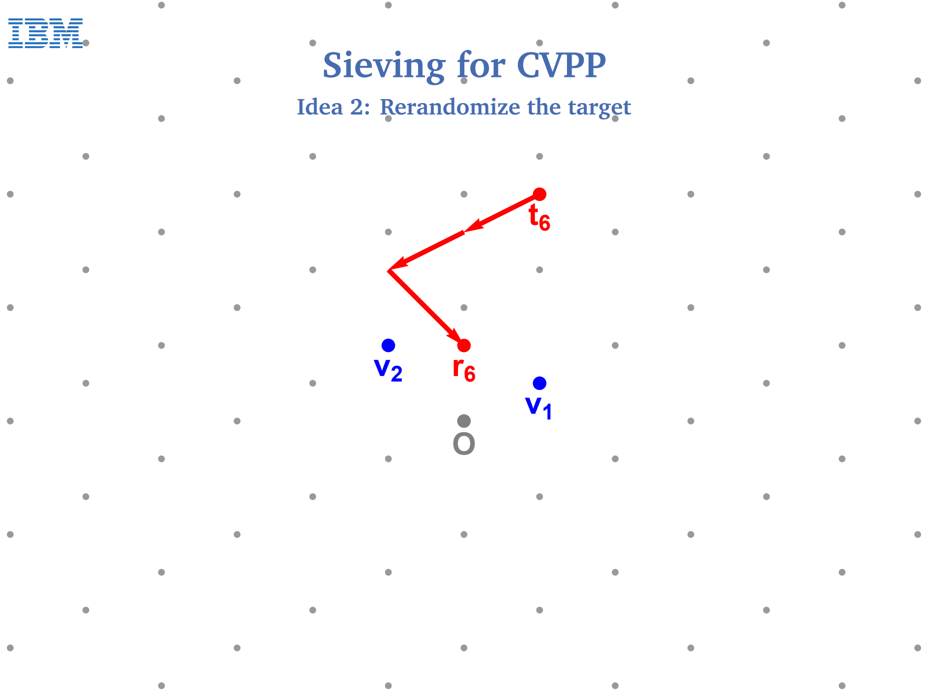
Sieving for CVPP

Idea 2: Rerandomize the target



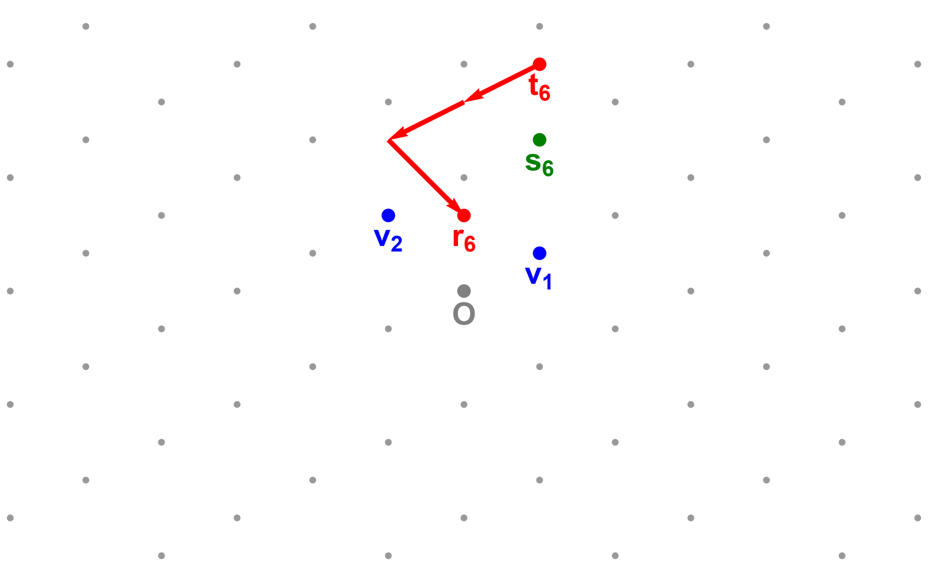
Sieving for CVPP

Idea 2: Rerandomize the target



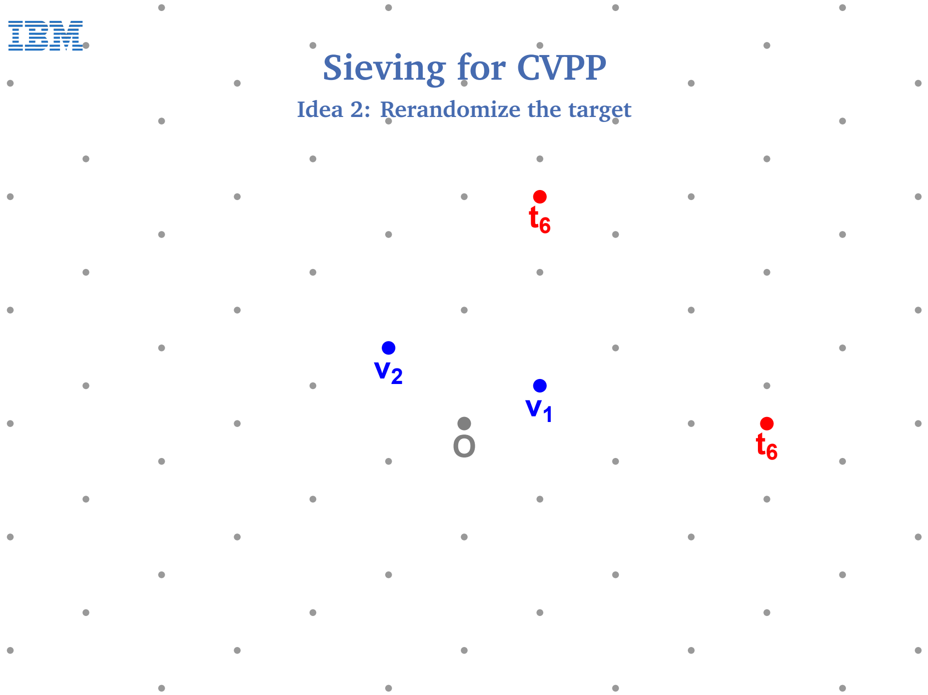
Sieving for CVPP

Idea 2: Rerandomize the target



Sieving for CVPP

Idea 2: Rerandomize the target

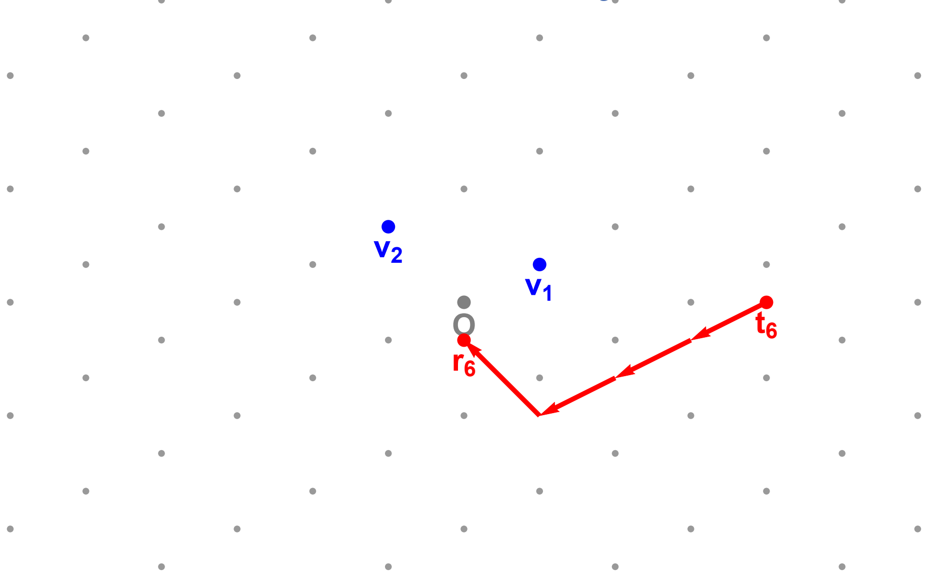


Sieving for CVPP

Idea 2: Rerandomize the target

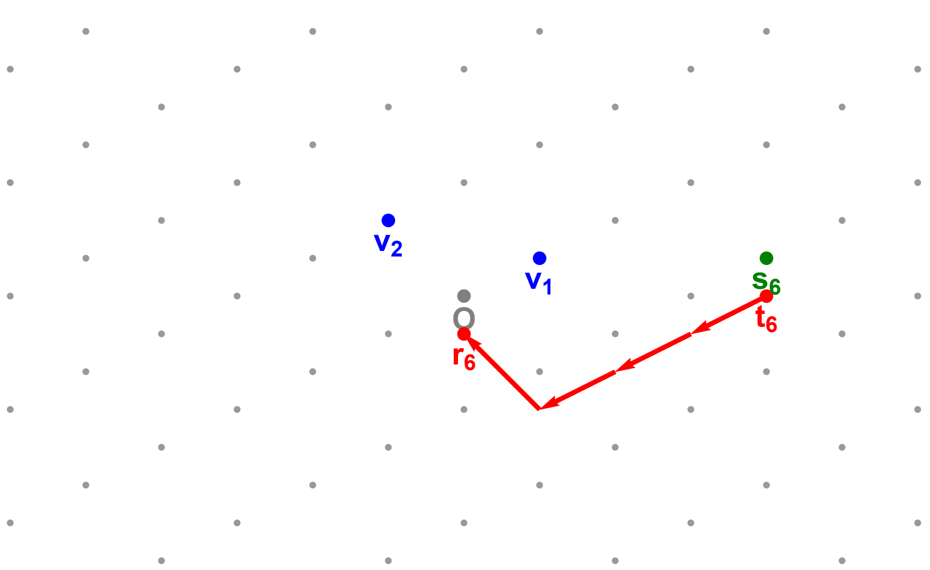
Sieving for CVPP

Idea 2: Rerandomize the target



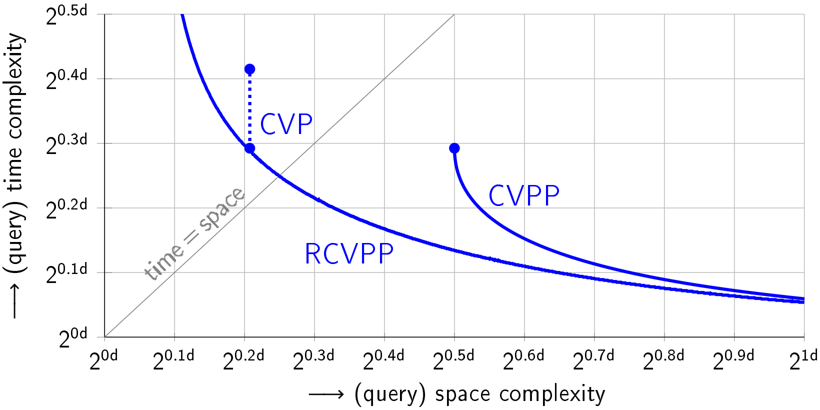
Sieving for CVPP

Idea 2: Rerandomize the target



Sieving for CVPP

Trade-offs



Conclusion

- Sieving for CVP similar costs as SVP

Conclusion

- Sieving for CVP similar costs as SVP
- Sieving for CVPP much easier than SVP

Conclusion

- Sieving for CVP similar costs as SVP
- Sieving for CVPP much easier than SVP
  - Preliminary experiments: 2000× faster in dimension 50

Conclusion

- Sieving for CVP similar costs as SVP
- Sieving for CVPP much easier than SVP
  - Preliminary experiments: $2000\times$ faster in dimension 50
  - Competitive with enumeration with pruning

Conclusion

- Sieving for CVP similar costs as SVP
- Sieving for CVPP much easier than SVP
  - Preliminary experiments: 2000× faster in dimension 50
  - Competitive with enumeration with pruning
- Better complexities for approximate CVP and BDD

Conclusion

- Sieving for CVP similar costs as SVP
- Sieving for CVPP much easier than SVP
  - Preliminary experiments: 2000× faster in dimension 50
  - Competitive with enumeration with pruning
- Better complexities for approximate CVP and BDD
- Open problem: hybrid enumeration with sieving

Conclusion

- Sieving for CVP similar costs as SVP
- Sieving for CVPP much easier than SVP
  - Preliminary experiments: 2000× faster in dimension 50
  - Competitive with enumeration with pruning
- Better complexities for approximate CVP and BDD
- Open problem: hybrid enumeration with sieving
  - Bottom part of enumeration tree corresponds to batch-CVP

Conclusion

- Sieving for CVP similar costs as SVP
- Sieving for CVPP much easier than SVP
  - Preliminary experiments: 2000× faster in dimension 50
  - Competitive with enumeration with pruning
- Better complexities for approximate CVP and BDD
- Open problem: hybrid enumeration with sieving
  - Bottom part of enumeration tree corresponds to batch-CVP
  - An efficient CVPP algorithm would speed up enumeration

Conclusion

- Sieving for CVP similar costs as SVP
- Sieving for CVPP much easier than SVP
  - Preliminary experiments: 2000× faster in dimension 50
  - Competitive with enumeration with pruning
- Better complexities for approximate CVP and BDD
- Open problem: hybrid enumeration with sieving
  - Bottom part of enumeration tree corresponds to batch-CVP
  - An efficient CVPP algorithm would speed up enumeration
  - CVPP in low dimension $\implies$ no memory issues

Questions?